PROGRAMMABLE LOGIC
DECEMBER 2013
ISSUE 281

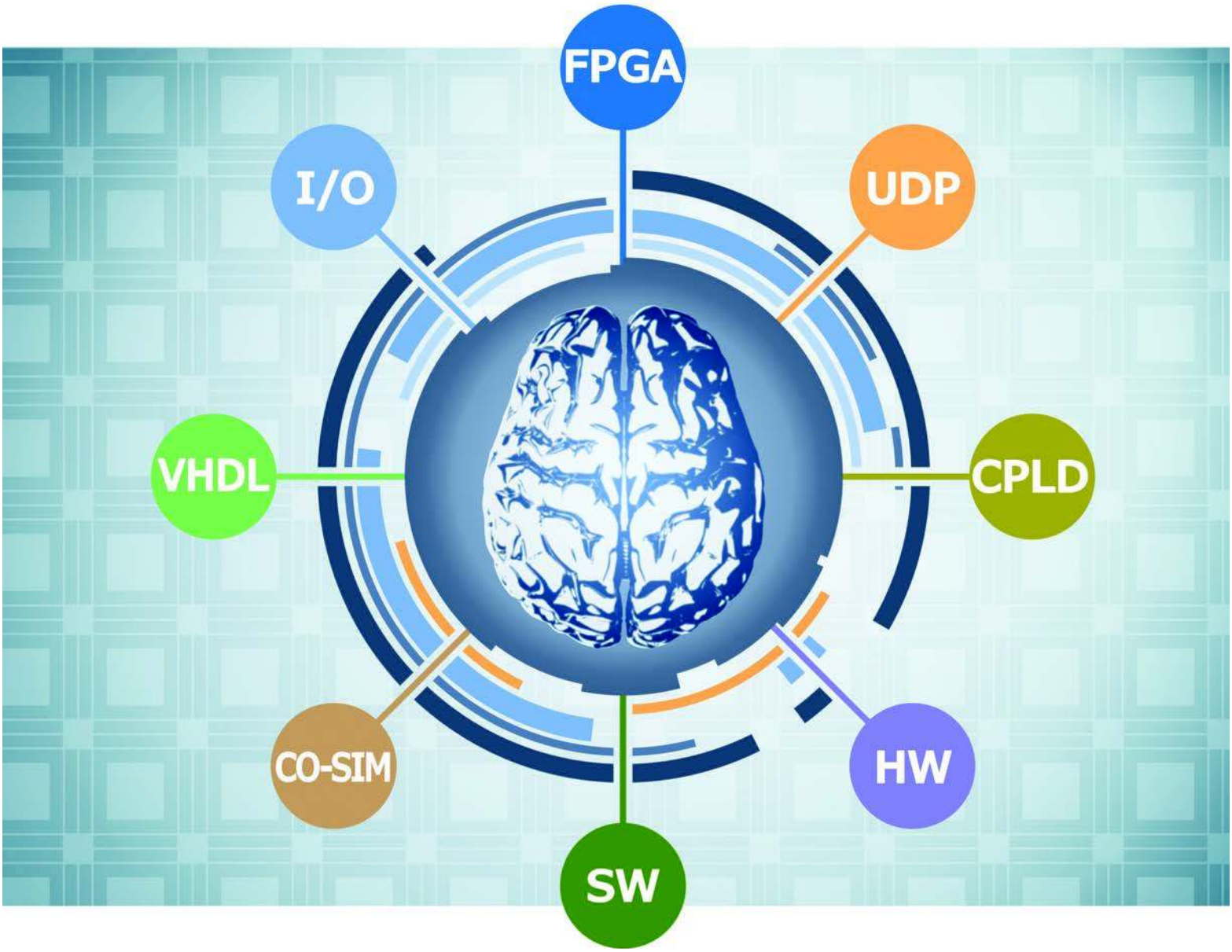
circuitcellar.com

circuit cellar

READY TO RECONFIGURE?

TIPS FOR WORKING CONFIDENTLY WITH FPGAs

Hardware Co-Simulation, Ethernet on an FPGA, & More



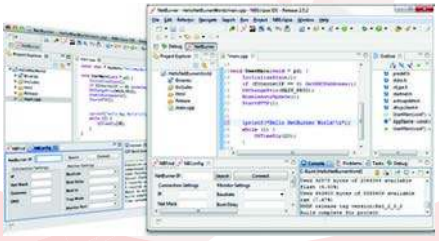
$9.00US $10.00CAN
12
0 74470 75349 0

■ SoC and IoT Predictions | Q&A: Smart-Camera Specialist
■ UDP Streaming | Converting A/D Values | New Memory Technologies
■ Impedance Matching | Embedded File Protection| HCS Testing |
Amplifier Class Comparison | USB Android Host IC ■ SBCs and STEM

Ethernet Core Modules with
High-Performance Connectivity Options

- **MOD5270**
  147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO
  3 UARTs · I²C · SPI
- **MOD5234**
  147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs
  I²C · SPI · CAN · eTPU (for I/O handling, serial communications, motor/timing/engine control applications)
- **MOD54415**
  250 MHz processor with 32MB flash & 64MB RAM · 42 GPIO · 8 UARTs
  5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire® interface
- **NANO54415**
  250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs
  4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire® interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor



The goal: Control, configure, or monitor a device using Ethernet

The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.

The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

MOD5270-100IR.....$69 (qty. 100)
MOD5234-100IR.....$99 (qty. 100)
MOD54415-100IR.....$89 (qty. 100)
NANO54415-200IR...$69 (qty. 100)

NNDK-MOD5270LC-KIT.....$99
NNDK-MOD5234LC-KIT.....$249
NNDK-MOD54415LC-KIT.....$129
NNDK-NANO54415-KIT.....$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

- **For additional information please visit**
  http://www.netburner.com/kits

NetBurner
Networking in One Day!

Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828

freescale™
Alliance Member

Issue 281 December 2013 | ISSN 1528-0608

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by:

Circuit Cellar, Inc.
111 Founders Plaza, Suite 300
East Hartford, CT 06108

Periodical rates paid at East Hartford, CT, and additional offices. One-year (12 issues) subscription rate US and possessions $50, Canada $65, Foreign/ ROW $75. All subscription orders payable in US funds only via Visa, MasterCard, international postal money order, or check drawn on US bank.

SUBSCRIPTIONS

Circuit Cellar, P.O. Box 462256, Escondido, CA 92046

E-mail: circuitcellar@pcspublink.com

Phone: 800.269.6301

Internet: circuitcellar.com

Address Changes/Problems: circuitcellar@pcspublink.com

Postmaster: Send address changes to
Circuit Cellar, P.O. Box 462256, Escondido, CA 92046

ADVERTISING

Strategic Media Marketing, Inc.
2 Main Street, Gloucester, MA 01930 USA

Phone: 978.281.7708

Fax: 978.281.7706

E-mail: circuitcellar@smmarketing.us
Advertising rates and terms available on request.

New Products:
New Products, Circuit Cellar, 111 Founders Plaza, Suite 300
East Hartford, CT 06108, E-mail: newproducts@circuitcellar.com

HEAD OFFICE

Circuit Cellar, Inc. 111 Founders Plaza, Suite 300
East Hartford, CT 06108
Phone: 860.289.0800

COVER PHOTOGRAPHY

Chris Rakoczy, www.rakoczyphoto.com

OVERCOME FEAR OF ETHERNET ON AN FPGA

As its name suggests, the appeal of an FPGA is that it is fully programmable. Instead of writing software, you design hardware blocks to quickly do what's required of a digital design. This also enables you to reprogram an FPGA product in the field to fix problems "on the fly."

But what if "you" are an individual electronics DIYer rather than an industrial designer? DIYers can find FPGAs daunting.

This *Circuit Cellar* issue should offer reassurance, at least on the topic of "UDP Streaming on an FPGA." That's the focus of Steffen Mauch's article for our Programmable Logic issue (p. 20).

Ethernet on an FPGA has several applications. For example, it can be used to stream measured signals to a computer for analysis or to connect a camera (via Camera Link) to an FPGA to transmit images to a computer.

Nonetheless, Mauch says, "most novices who start to develop FPGA solutions are afraid to use Ethernet or DDR-SDRAM on their boards because they fear the resulting complexity." Also, DIYers don't have the necessary IP core licenses, which are costly and often carry restrictions.

Mauch's UDP monitor project avoids such costs and restrictions by using a free implementation of an Ethernet-streaming device based on a Xilinx Spartan-6 LX FPGA. His article explains how to use OpenCores's open-source tri-mode MAC implementation and stream UDP packets with VHDL over Ethernet.

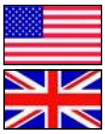
Mauch is not the only writer offering insights into FPGAs. For more advanced FPGA enthusiasts, columnist Colin O'Flynn discusses hardware co-simulation (HCS), which enables the software simulation of a design to be offloaded to an FPGA. This approach significantly shortens the time needed for adequate simulation of a new product and ensures that a design is actually working in hardware (p. 52).

This *Circuit Cellar* issue offers a number of interesting topics in addition to programmable logic. For example, you'll find a comprehensive overview of the latest in memory technologies, advice on choosing a flash file system for your embedded Linux system, a comparison of amplifier classes, and much more.

Mary Wilson

editor@circuitcellar.com

THE TEAM



USA/UK
Carlo van Nistelrooy
+1 860 289 0800
c.vannistelrooy@elektor.com

ELEKTOR LABS
Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

GERMANY
Ferdinand te Walvaart
+49 241 88 909-17
f.tewalvaart@elektor.de

FRANCE
Denis Meyer
+31 46 4389435
d.meyer@elektor.fr

NETHERLANDS
Harry Baggen
+31 46 4389429
h.baggen@elektor.nl

SPAIN
Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es

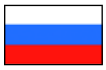
ITALY
Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it

SWEDEN
Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

BRAZIL
João Martins
+31 46 4389444
j.martins@elektor.com

PORTUGAL
João Martins
+31 46 4389444
j.martins@elektor.com

INDIA
Sunil D. Malekar
+91 9833168815
ts@elektor.in



RUSSIA
Nataliya Melnikova
+7 965 395 33 36
elektor.russia@gmail.com

TURKEY
Zeynep Köksal
+90 532 277 48 26
zkoksal@beti.com.tr

SOUTH AFRICA
Johan Dijk
+31 6 1589 4245
j.dijk@elektor.com

CHINA
Cees Baay
+86 21 6445 2811
ceesbaay@gmail.com

OUR NETWORK

circuit cellar

VOICE COIL

audioxpress

elektor

SUPPORTING COMPANIES

NOT A SUPPORTING COMPANY YET?

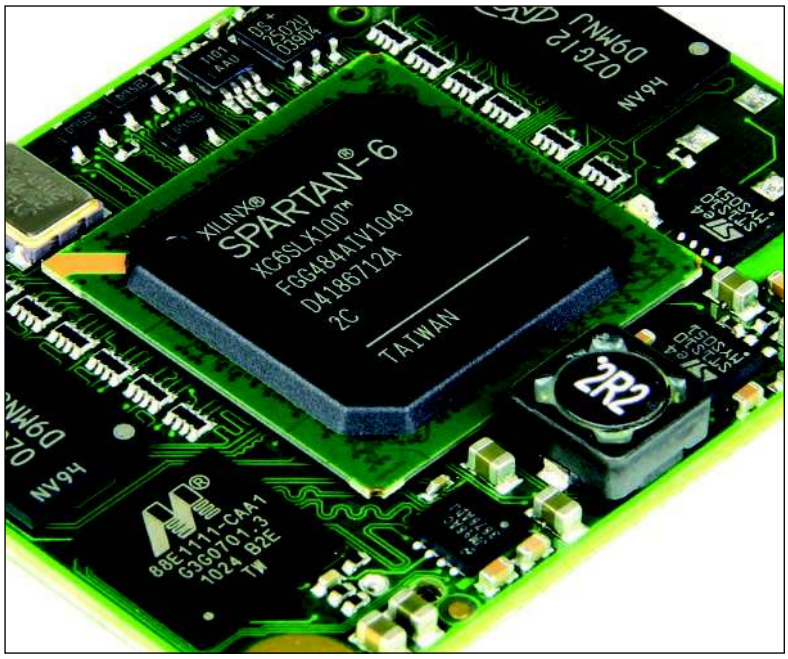
Contact Peter Wostrel (circuitcellar@smmarketing.us, Phone 978.281.7708, Fax 978.281.7706) to reserve your own space for the next edition of our members' magazine.

CONTENTS

circuit cellar

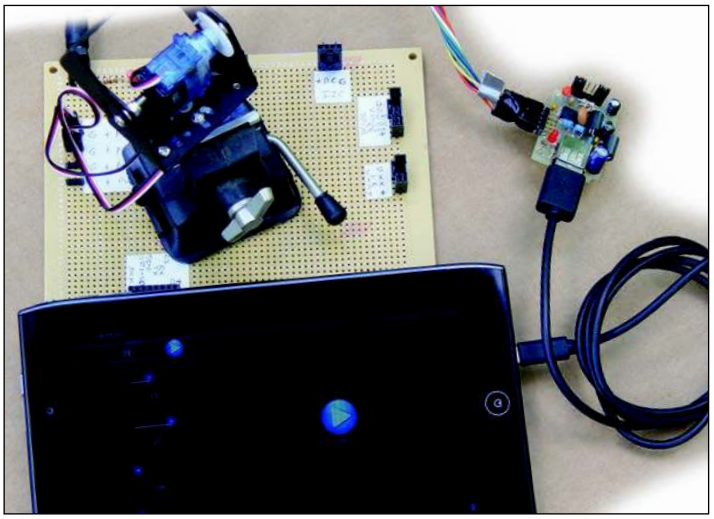
DECEMBER 2013 • ISSUE 281

PROGRAMMABLE LOGIC

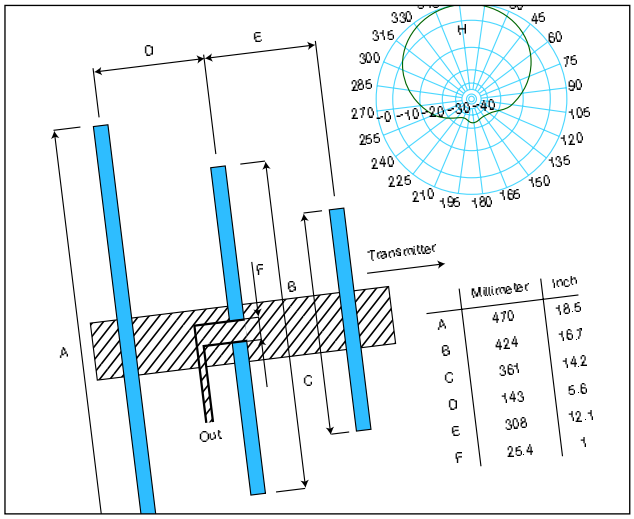


UDP STREAMING DEVICE BASED ON AN FPGA MICROMODULE

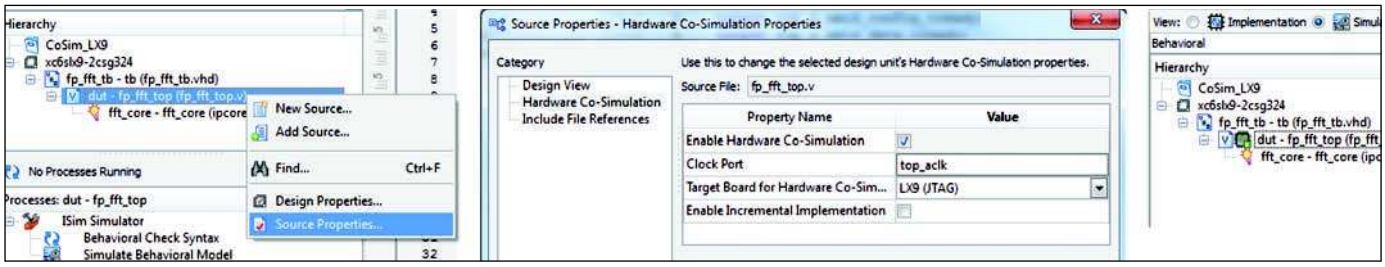
CONTENTS



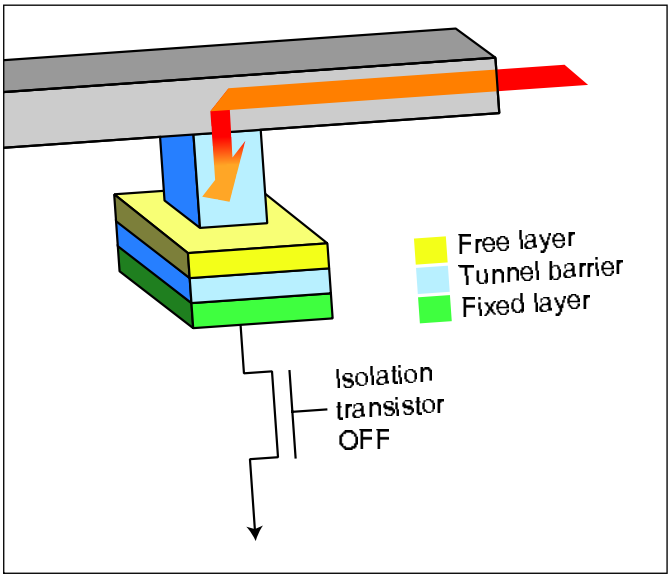
SERVO APP



DIPOLE ANTENNA FOR 345 MHz



ENABLING HARDWARE CO-SIMULATION

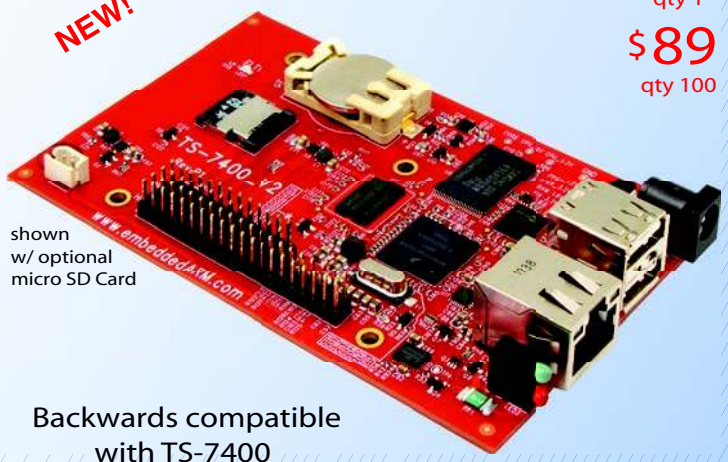


SPIN-TORQUE MRAM CELL'S STRUCTURE

Low Power
Low Cost
High Reliability

pricing starts at
$119 qty 1
$89 qty 100

NEW!



shown w/ optional micro SD Card

Backwards compatible with TS-7400

Single Board Computers

TS-7400-V2 450MHz ARM9

Features:

- Up to 256MB RAM
- 2GB NAND Flash
- 2x CAN port
- 20x DIO
- 1x Watchdog Timer
- 2x SD Sockets
- 1x BBRTC
- 1x Temp Sensor

Benefits:

- -40 to +85C Industrial temperature range
- Backward compatible w/ TS-7400
- 2x faster, more features, reduced cost
- Low power with 120mW standby mode
- Boots quickly to your Embedded Application
- Easy development w/ Debian and Linux 2.6

Computer-on-Modules

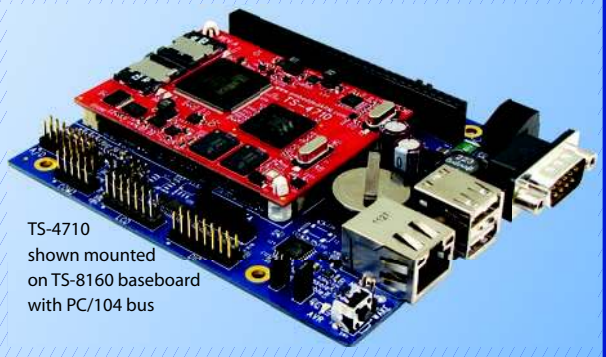
State of the Art Embedded Design

Our embedded Computer-on-Modules (COMs) are CPU core modules that securely connect to a carrier board. Standard functions are designed into the Computer-on-Module which is mated to your custom carrier board to enable exactly the functions your project requires. COTS carrier boards are available, or design your own carrier board for a custom solution with reduced design time and complexity.

Features can include:

- Up to 1GHz ARM w/ 512MB RAM
- Fanless, low power designs
- All parts soldered on, no moving parts
- Double Store SD for reliable storage
- User-programmable FPGAs
- -40 to +85C Industrial temperature range
- Easy development w/ Debian and Linux 2.6
- Sub-second boot to Linux

pricing starts at $189 qty 1 $138 qty 100



TS-4710 shown mounted on TS-8160 baseboard with PC/104 bus

Computer-on-Modules include:

- TS-4200: Atmel ARM9 w/ super low power
- NEW! TS-4600: 400MHZ at extremely low cost
- TS-4710: Up to 1066MHz PXA168 w/ video
- NEW! TS-4712: like TS-4710 + 2 Ethernets
- NEW! TS-4720: like TS-4712 + 4GB eMMC Flash
- TS-4800: 800MHz FreeScale iMX515 w/ video

- 30 years in business
- Open Source Vision
- 8+ Year Product Lifecycle Guarantee
- Never discontinued a product
- Next day shipping for most products
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time

Design your solution with one of our engineers (480) 837-5200

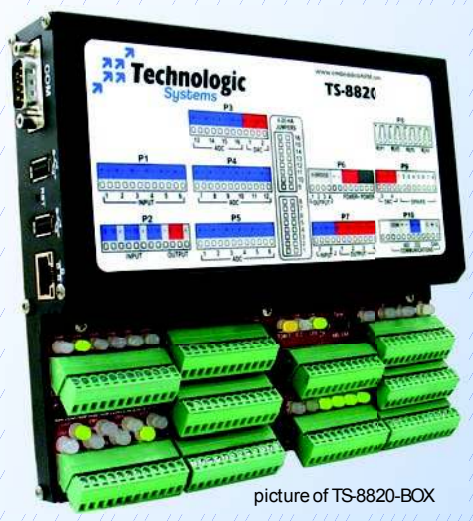
Touch Panel Computers
Panel Mount or Fully Enclosed

pricing starts at $409 qty 1 $369 qty 100

Features can include:

- 5”-inch, 7-inch and 10-inch touchscreens
- Fanless operation from -20 to +70C
- 800MHz or 1066MHz ARM9 CPU
- Up to 512MB RAM, 256MB SLC XNAND
- 2x microSD with DoubleStore
- 2x Ethernet, 2x USB Host
- CAN, RS-232, SPI, I2C, DIO
- 1x RS485 2W-Modbus
- Optional cellular, WIFI & XBEE radios
- Linux 2.6.34 w/ sub-second boot
- Headphone connector & speaker

TS-TPC-8380 fully enclosed in low cost plastic enclosure
Panel Mount (Open Frame) TPCs also available



picture of TS-8820-BOX

Controllers start at $199 qty 1

Modbus Peripherals start at $119 qty 100

Industrial Controllers

- 250MHz to 1GHz (CPU
- Fast startup (under 3 seconds)
- Opto-isolated ports available
- Industrial screw-down connectors
- Digital counters, quadrature decoder
- Flexible programming with Debian Linux

Modbus Peripherals

- Support 2W-MODBUS communication and power over a single CAT5 cable

NEW!
- TS-1420 12 outputs 8Amp
- TS-1700 DIO and One -Wire

NEW!
- TS-1800 14 A/D, 24 DIO
- TS-1400 6 relays

Technologic Systems now offers four powerful computers targeting industrial process control. Implement an intelligent automation system at low cost with a minimal number of components.

Technologic Systems

Visit our TS-7800 powered website at

www.embeddedARM.com

CC WORLD

SoC CONFERENCE NOTES

by CC Staff (USA)

The 11th International System on a Chip (SoC) Conference comprised two days of SoC-related sessions, meetings, and keynotes at the University of California, Irvine. Topics ranged from emerging complex SOCs to Internet of Things (IoT) security to analog and mixed-signal SoC solutions. Below are conference notes that embedded systems designers will find useful.

Calit2, University of California, Irvine

- **"Living on the Edge (of the SoC)," Jim Aralis, CTO and VP of R&D at Microsemi**
  - Industry-Related Observations:
    - Right now we're seeing a decline in standard products. "Manufacturers of standard products have to look to a different future, and they are." The result? Companies like Microsemi are "absorbing some of what our customer does."
  - Predictions:
    - Expertise in analog processing, signal conditioning, precision timing, and high-speed communications design will be essential for future engineers to develop successful products in an ever-changing digital environment.
    - System engineers and architects will be the "creators" and the companies with the best system engineers will excel. This will likely result in the reduced need for board designers.
    - We'll see fewer foundries, fewer fabless design houses, fewer SoC designers, digital will become block based, and "system houses will take over much of the differentiation," Aralis said. "Analog will continue to drive some of that differentiation… It's going to be an interesting new world. And I think it's going to be very good, just make sure you are in the right place."

- **"Securing the 'Internet of Things,'" Jauher Zaidi, Chairman & CTO, Palmchip**
  - Industry-Related Observations:
    - Zaidi noted that Cisco IBSG reported that there will be 50 billion devices connected to the Internet by 2020 (D. Evans, "The Internet of Things," Cisco IBSG, 2011). He also noted that by 2020, the average person will be connected to approximately 5,000 devices.
    - The proliferation of Internet-connected devices means new security issues for end users and manufacturers.
  - Secure Design Planning:
    - Engineers must understand that personal privacy will be a top concern of consumers in the next decade and beyond.
    - "Security has to be a part of a design," Zaidi said. This means more than focusing on data storage and transmission-related security issues. Engineers must also focus on chip security.

Send CC your workspace pics & info

E-mail your photos and info to editor@circuitcellar.com

Tweet pics (or a link) of your space to @circuitcellar. Use #ccworkspaces

WORKSPACE FOR DEV SYSTEMS

by CC Staff (USA)

The CC editorial team continues reviewing engineering workspaces each month. Currently, we're featuring embedded systems engineer David Cass Tyler's New Mexico-based home setup.

"When I require extra space to spread out, I move into the spare bedroom and use the desk in there to set up the hardware," Tyler noted. "Almost all of my projects are developed to be distributed and accessible through the network. When I need to program on a different computer, I tend to use the remote desktop to program on other Windows-based systems. There is seldom a time when I have to physically move to one of the other systems."

We'd like to know about your workspace. Send our editors your photos and information!

Need to Stay Connected?

Learn About Microchip's USB and Networking Solutions

USB

USB technology can be found in practically all applications and markets—from consumer, industrial and automotive segments. The extreme proliferation of USB has even led to the adoption of the technology as an embedded chip-to-chip high bandwidth interface. Microchip has been enabling such uses, applications and markets with seamless USB connectivity by delivering integrated value rich solutions, such as USB hub controllers, power delivery and charging, transceivers/switches, Flash media controllers and security.

Ethernet

Ethernet devices have become ubiquitous in communications and networking products, servicing a wide variety of applications across multiple market segments. This well understood technology provides a robust link to ensure reliable communication between devices in a network. Microchip has a broad portfolio of reliable, high quality and high performance Ethernet solutions. From Ethernet Switches, Controllers, Bridges, and PHYs, to a variety of standard interfaces, serving consumer, industrial, and automotive applications, Microchip can provide a solution to address your varied application needs and offers you the support needed to reduce your time to market.

GETTING STARTED IS EASY

If you need USB or Networking connectivity in your product design, Microchip's broad line of proven and reliable solutions cover all main market segments and applications including:

- Consumer electronics
- Mobile
- Industrial equipment and control
- Automotive
- Medical

Visit www.microchip.com/connectivity for more information.

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

QUESTIONS & ANSWERS

Embedded Computing Expert

An Interview with Marilyn Wolf

Marilyn Wolf has created embedded computing techniques, co-founded two companies, and received several Institute of Electrical and Electronics Engineers (IEEE) distinctions. She is currently teaching at Georgia Institute of Technology's School of Electrical and Computer Engineering and researching smart-energy grids.—Nan Price, Associate Editor

NAN: Do you remember your first computer engineering project?

MARILYN: My dad is an inventor. One of his stories was about using copper sewer pipe as a drum memory. In elementary school, my friend and I tried to build a computer and bought a PCB fabrication kit from RadioShack. We carefully made the switch features using masking tape and etched the board. Then we tried to solder it and found that our patterning technology outpaced our soldering technology.

NAN: You have developed many embedded computing techniques—from hardware/software co-design algorithms and real-time scheduling algorithms to distributed smart cameras and code compression. Can you provide some information about these techniques?

MARILYN: I was inspired to work on co-design by my boss at Bell Labs, Al Dunlop. I was working on very-large-scale integration (VLSI) CAD at the time and he brought in someone who designed consumer telephones. Those designers didn't care a bit about our fancy VLSI because it was too expensive. They wanted help designing software for microprocessors.

Microprocessors in the 1980s were pretty small, so I started on simple problems, such as partitioning a specification into software plus a hardware accelerator. Around the turn of the millennium, we started to see some very powerful processors (e.g., the Philips Trimedia). I decided to pick up on one of my earliest interests, photography, and look at smart cameras for real-time computer vision.

That work eventually led us to form Verificon, which developed smart camera systems. We closed the company because the market for surveillance systems is very competitive.

We have started a new company, SVT Analytics, to pursue customer analytics for retail using smart camera technologies. I also continued to look at methodologies and tools for bigger software systems, yet another interest I inherited from my dad.

NAN: Tell us a little more about SVT Analytics. What services does the company provide and how does it utilize smart-camera technology?

MARILYN: We started SVT Analytics to develop customer analytics for software. Our goal is to do for bricks-and-mortar retailers what web retailers can do to learn about their customers.

On the web, retailers can track the pages customers visit, how long they stay at a page, what page they visit next, and all sorts of other statistics. Retailers use that information to suggest other things to buy, for example.

Bricks-and-mortar stores know what sells but they don't know why. Using computer vision, we can determine how long people stay in a particular area of the store, where they came from, where they go to, or whether employees are interacting with customers.

Our experience with embedded computer vision helps us develop algorithms that are accurate but also run on inexpensive

platforms. Bad data leads to bad decisions, but these systems need to be inexpensive enough to be sprinkled all around the store so they can capture a lot of data.

NAN: Can you provide a more detailed overview of the impact of IC technology on surveillance in recent years? What do you see as the most active areas for research and advancements in this field?

MARILYN: Moore's law has advanced to the point that we can provide a huge amount of computational power on a single chip. We explored two different architectures: an FPGA accelerator with a CPU and a programmable video processor.

We were able to provide highly accurate computer vision on inexpensive platforms, about $500 per channel. Even so, we had to design our algorithms very carefully to make the best use of the compute horsepower available to us.

Computer vision can soak up as much computation as you can throw at it. Over the years, we have developed some secret sauce for reducing computational cost while maintaining sufficient accuracy.

NAN: You wrote several books, including *Computers as Components: Principles of Embedded Computing System Design* and *Embedded Software Design and Programming of Multiprocessor System-on-Chip: Simulink and System C Case Studies*. What can readers expect to gain from reading your books?

MARILYN: *Computers as Components* is an undergraduate text. I tried to hit the fundamentals (e.g., real-time scheduling theory, software performance analysis, and low-power computing) but wrap around real-world examples and systems.

Embedded Software Design is a research monograph that primarily came out of Katalin Popovici's work in Ahmed Jerraya's group. Ahmed is an old friend and collaborator.

NAN: When did you transition from engineering to teaching? What prompted this change?

MARILYN: Actually, being a professor and teaching in a classroom have surprisingly little to do with each other. I spend a lot of time funding research, writing proposals, and dealing with students.

I spent five years at Bell Labs before moving to Princeton, NJ. I thought moving to a new environment would challenge me, which is always good. And although we were very well supported at Bell Labs, ultimately we had only one customer for our ideas. At a university, you can shop around to find someone interested in what you want to do.

NAN: How long have you been at Georgia Institute of Technology's School of Electrical and Computer Engineering? What courses do you currently teach and what do you enjoy most about instructing?

MARILYN: I recently designed a new course, Physics of Computing, which is a very different take on an introduction to computer engineering. Instead of directly focusing on logic design and computer organization, we discuss the physical basis of delay and energy consumption.

You can talk about an amazingly large number of problems involving just inverters and RC circuits. We relate these basic physical phenomena to systems. For example, we figure out why dynamic RAM (DRAM) gets bigger but not faster, then see

"Moore's law has advanced to the point that we can provide a huge amount of computational power on a single chip."

QUESTIONS & ANSWERS

how that has driven computer architecture as DRAM has hit the memory wall.

NAN: As an engineering professor, you have some insight into what excites future engineers. With respect to electrical engineering and embedded design/programming, what are some "hot topics" your students are currently attracted to?

MARILYN: Embedded software—real-time, low-power—is everywhere. The more general term today is "cyber-physical systems," which are systems that interact with the physical world. I am moving slowly into control-oriented software from signal/image processing. Closing the loop in a control system makes things very interesting.

My Georgia Tech colleague Eric Feron and I have a small project on jet engine control. His engine test room has a 6" thick blast window. You don't get much more exciting than that.

"Smart-energy grids are a prime example of Internet-based control. But the Internet isn't very good at real-time control."

NAN: That does sound exciting. Tell us more about the project and what you are exploring with it in terms of embedded software and closed-loop control systems.

MARILYN: Jet engine designers are under the same pressures now that have faced car engine designers for years: better fuel efficiency, lower emissions, lower maintenance cost, and lower noise. In the car world, CPU-based engine controllers were the critical factor that enabled car manufacturers to simultaneously improve fuel efficiency and reduce emissions.

Jet engines need to incorporate more sensors and more computers to use those sensors to crunch the data in real time and figure out how to control the engine. Jet engine designers are also looking at more complex engine designs with more flaps and controls to make the best use of that sensor data.

One challenge of jet engines is the high temperatures. Jet engines are so hot that some parts of the engine would melt without careful design. We need to provide more computational power while living with the restrictions of high-temperature electronics.

NAN: Your research interests include embedded computing, smart devices, VLSI systems, and biochips. What types of projects are you currently working on?

MARILYN: I'm working on with Santiago Grivalga of Georgia Tech on smart-energy grids, which are really huge systems that would span entire countries or continents. I continue to work on VLSI-related topics, such as the work on error-aware computing that I pursued with Saibal Mukopodhyay.

I also work with my friend Shuvra Bhattacharyya on architectures for signal-processing systems. As for more unusual things, I'm working on a medical device project that is at the early stages, so I can't say too much specifically about it.

NAN: Can you provide more specifics about your research into smart energy grids?

MARILYN: Smart-energy grids are also driven by the push for greater efficiency. In addition, renewable energy sources have different characteristics than traditional coal-fired generators. For example, because winds are so variable, the energy produced by wind generators can quickly change.

The uses of electricity are also more complex, and we see increasing opportunities to shift demand to level out generation needs. For example, electric cars need to be recharged, but that can happen during off-peak hours. But energy systems are huge. A single grid covers the eastern US from Florida to Minnesota.

To make all these improvements requires sophisticated software and careful design to ensure that the grid is highly reliable. Smart-energy grids are a prime example of Internet-based control.

We have so many devices on the grid that need to coordinate that the Internet is the only way to connect them. But the Internet isn't very good at real-time control, so we have to be careful.

We also have to worry about security Internet-enabled devices enable smart grid operations but they also provide opportunities for tampering.

NAN: You've earned several distinctions. You were the recipient of the Institute of Electrical and Electronics Engineers (IEEE) Circuits and Systems Society Education Award and the IEEE Computer Society Golden Core Award. Tell us about these experiences.

MARILYN: These awards are presented at conferences. The presentation is a very warm, happy experience. Everyone is happy. These things are time to celebrate the field and the many friends I've made through my work.

Hard-Wired.

At Newark element14, we connect you to all of your cable and wire management needs. **newark.com**

3M AlphaWire Amphenol BELDEN BRADY KROY
General Cable CAROL BRAND PANDUIT pro-POWER TE connectivity Thomas&Betts

Newark element14

MEMBER PROFILE

Walter O. Krawec

Research Assistant and PhD Student, Stevens Institute of Technology, Hoboken, NJ • Upstate New York

MEMBER STATUS:

Walter has been reading *Circuit Cellar* since he got his first issue in 1999. Free copies were available at the Trinity College Fire Fighting Robot Contest, which was his first experience with robotics. *Circuit Cellar* was the first magazine for which he wrote an article ("An HC11 File Manager," two-part series, issues 129 and 130, 2001).

TECH INTERESTS:

Robotics, among other things. He is particularly interested in developmental and evolutionary robotics (where the robot's strategies, controllers, and so forth are evolved instead of programmed in directly).

RECENT TECH ACQUISITION:

Walter is enjoying his Raspberry Pi. "What a remarkable product! I think it's great that I can take my AI software, which I've been writing on a PC, copy it to the Raspberry Pi, compile it with GCC, then off it goes with little or no modification!"

CURRENT PROJECTS:

Walter is designing a new programming language and interpreter (for Windows/Mac/Linux, including the Raspberry Pi) that uses a simulated quantum computer to drive a robot. "What better way to learn the basics of quantum computing than by building a robot around one?" The first version of this language is available on his website (walterkrawec.org). He has plans to release an improved version.

THOUGHTS ON EMBEDDED TECH:

Walter said he is amazed with the power of the latest embedded technology, for example the Raspberry Pi. "For less than $40 you have a perfect controller for a robot that can handle incredibly complex programs. Slap on one of those USB battery packs and you have a fully mobile robot," he said. He used a Pololu Maestro to interface the motors and analog sensors. "It all works and it does everything I need." However, he added, "If you want to build any of this yourself by hand it can be much harder, especially since most of the cool stuff is surface mount, making it difficult to get started."

Outstanding speed optimizations enable IAR Embedded Workbench to generate faster code than ever before. With the shortest possible execution times it is the ultimate choice for developing low-power applications. The world-leading C/C++ compiler and debugger tool suite, with the broadest MCU support, is now even more powerful.

www.iar.com

IAR SYSTEMS

PRODUCT NEWS

SMALL CELLULAR EMBEDDED PLUG-IN MODEM

The **Skywire** is a small embedded plug-in cellular modem that uses a standard XBEE form factor and 1xRTT CDMA operating mode, which minimizes hardware and network costs. Its U.FL port helps ensure antenna flexibility. The modem's bundled carrier service plans enable simple deployment.

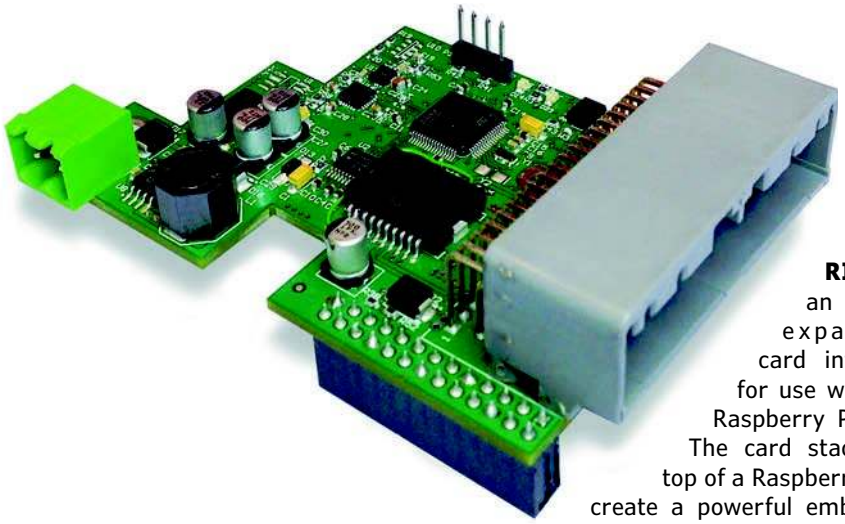
The Skywire embedded modem features a Telit CE910-DUAL wireless module and is available with bundled CDMA 1xRTT data plans. This enables developers to add fully compliant cellular connectivity without applying for certification. Future versions of the Skywire will support GSM and LTE.

The Skywire is available with a complete development kit that includes the cellular modem, a baseboard, an antenna, a power supply, debug cables, and a cellular service plan. The Skywire baseboard is an Arduino shield that enables direct connection to an Arduino microcontroller.

Skywire modems cost **$129** for one and **$99** in 1,000-unit quantities. A complete development kit including the modem costs **$262**.

NimbeLink, LLC
www.nimbelink.com

I/O RASPBERRY PI EXPANSION CARD



The **RIO** is an I/O expansion card intended for use with the Raspberry Pi SBC. The card stacks on top of a Raspberry Pi to create a powerful embedded control and navigation computer in a small 20-mm × 65-mm × 85-mm footprint. The RIO is well suited for applications requiring real-world interfacing, such as robotics, industrial and home automation, and data acquisition and control.

The RIO adds 13 inputs that can be configured as digital inputs, 0-to-5-V analog inputs with 12-bit resolution, or pulse inputs capable of pulse width, duty cycle, or frequency capture. Eight digital outputs are provided to drive loads up to 1 A each at up to 24 V.

The RIO includes a 32-bit ARM Cortex M4 microcontroller that processes and buffers the I/O and creates a seamless communication with the Raspberry Pi. The RIO processor can be user-programmed with a simple BASIC-like programming language, enabling it to perform logic, conditioning, and other I/O processing in real time. On the Linux side, RIO comes with drivers and a function library to quickly configure and access the I/O and to exchange data with the Raspberry Pi.

The RIO features several communication interfaces, including an RS-232 serial port to connect to standard serial devices, a TTL serial port to connect to Arduino and other microcontrollers that aren't equipped with a RS-232 transceiver, and a CAN bus interface.

The RIO is available in two versions. The RIO-BASIC costs **$85** and the RIO-AHRS costs **$175**.

Roboteq, Inc.
www.roboteq.com

IoT ASAP

The easiest way to add wireless is now the easiest way to connect to the Internet of Things

The new XBee® Wi-Fi Cloud Kit from Digi International brings the Internet of Things to the popular XBee platform. Built around Digi's new XBee Wi-Fi module, which fully integrates into Device Cloud, the kit is a simple way for anyone with an interest in M2M and the Internet of Things to build a hardware prototype and integrate it into an internet-based application. This kit is suitable for electronics engineers, software designers, educators, makers and innovators of all kinds.

- The fastest and easiest way to explore and develop Internet-connected devices and circuits
- Free, open-source application enables two-way communication and control with the development board over the internet
- Fully customizable widgets for simple monitoring and control of connected devices
- Free limited subscription to Device Cloud by Etherios™

GET STARTED $149.00

XBee Wi-Fi Cloud Kit

Kit contains an XBee Wi-Fi module; a development board with a variety of sensors and actuators; a variety of loose electronic prototyping parts for making circuits of your own; and cables, accessories and everything needed to connect to the web (minus the AP).

digi.com/xbeewificloudkit

#IoTASAP

XBee® By Digi

DEVICE CLOUD by ETHERIOS

Your M2M Expert™

Digi®

PRODUCT NEWS

SMALL, SELF-CONTAINED GNSS RECEIVER

TM Series GNSS modules are self-contained, high-performance global navigation satellite system (GNSS) receivers designed for navigation, asset tracking, and positioning applications. Based on the MediaTek chipset, the receivers can simultaneously acquire and track several satellite constellations, including the US GPS, Europe's GALILEO, Russia's GLONASS, and Japan's QZSS.

The 10-mm × 10-mm receivers are capable of better than 2.5-m position accuracy. Hybrid ephemeris prediction can be used to achieve less than 15-s cold start times. The receiver can operate down to 3 V and has a 20-mA low tracking current. To save power, the TM Series GNSS modules have built-in receiver duty cycling that can be configured to periodically turn off. This feature, combined with the module's low power consumption, helps maximize battery life in battery-powered systems.

The receiver modules are easy to integrate, since they don't require software setup or configuration to power up and output position data. The TM Series GNSS receivers use a standard UART serial interface to send and receive NMEA messages in ASCII format. A serial command set can be used to configure optional features. Using a USB or RS-232 converter chip, the modules' UART can be directly connected to a microcontroller or a PC's UART.

The GPS Master Development System connects a TM Series Evaluation Module to a prototyping board with a color display that shows coordinates, a speedometer, and a compass for mobile evaluation. A USB interface enables simple viewing of satellite data and Internet mapping and custom software application development.

Contact Linx Technologies for pricing.

Linx Technologies
www.linxtechnologies.com

CLIENT PROFILE

Digi International, Inc.

www.digi.com

11001 Bren Road East, Minnetonka, MN 55343

CONTACT: Elizabeth Presson, elizabeth.presson@digi.com

FEATURED PRODUCT: The XBee product family (www.digi.com/xbee) is a series of modular products that make adding wireless technology easy and cost-effective. Whether you need a ZigBee module or a fast multipoint solution, 2.4 GHz or long-range 900 MHz—there's an XBee to meet your specific requirements.

PRODUCT INFORMATION: Digi now offers the XBee Wi-Fi Cloud Kit (www.digi.com/xbeewificloudkit) for those who want to try the XBee Wi-Fi (XB2B-WFUT-001) with seamless cloud connectivity. The Cloud Kit brings the Internet of Things (IoT) to the popular XBee platform. Built around Digi's new XBee Wi-Fi module, which fully integrates into the Device Cloud by Etherios, the kit is a simple way for anyone with an interest in M2M and the IoT to build a hardware prototype and integrate it into an Internet-based application. This kit is suitable for electronics engineers, software designers, educators, and innovators.

EXCLUSIVE OFFER: The XBee Wi-Fi Cloud Kit includes an XBee Wi-Fi module; a development board with a variety of sensors and actuators; loose electronic prototyping parts to make circuits of your own; a free subscription to Device Cloud; fully customizable widgets to monitor and control connected devices; an open-source application that enables two-way communication and control with the development board over the Internet; and cables, accessories, and everything needed to connect to the web. The Cloud Kit costs $149.

Circuit Cellar prides itself on presenting readers with information about innovative companies, organizations, products, and services relating to embedded technologies. This space is where Circuit Cellar enables clients to present readers useful information, special deals, and more.

INDUSTRY & ENTERPRISE

FEATURES

UDP Streaming on an FPGA

When implementing a high-performance network device with an FPGA, you don't always have to use a proprietary IP core. User Datagram Protocol streaming is another option—and it's not as difficult as it may seem. The article shows that a hardware implementation without a processor is a practical solution, even for novices.

By Steffen Mauch (Germany)

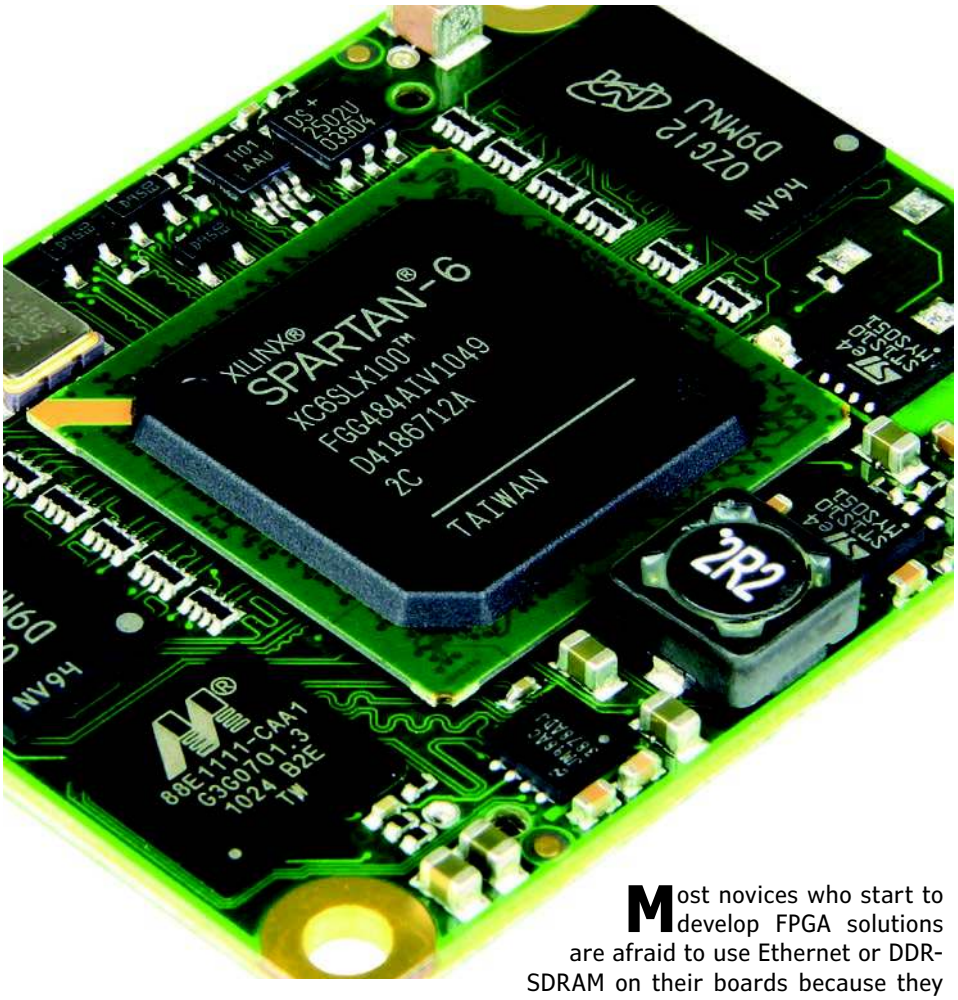


Photo 1
Trenz Electronic's TE0600 is a credit-card sized industrial FPGA micromodule that integrates a Xilinx Spartan-6 LX FPGA. (Photo courtesy of Trenz Electronics)

Most novices who start to develop FPGA solutions are afraid to use Ethernet or DDR-SDRAM on their boards because they fear the resulting complexity. Moreover, they don't have the necessary IP cores or the information. When implementing a high-performance network device with an FPGA, you typically use a proprietary IP core, usually from the FPGA's manufacturer. Most FPGA development novices are nervous when they need to integrate something apparently complex (e.g., network streaming into a FPGA).

Furthermore, licenses for IP cores, such as Xilinx's Tri-Mode Ethernet Media Access Controller (TEMAC) are costly. Other proprietary media access controllers (MACs), such as Xilinex's AXI Ethernet Lite MAC, cannot be used without Xilinx's Embedded Development Kit (EDK). They may also require licenses or have other restrictions.

For this project, I used a free implementation of an Ethernet streaming device based on Trenz Electronic's TE0600 FPGA micromodule (see **Photo 1**). The TE0600 is based on a Xilinx Spartan-6 LX FPGA. Different FPGA sizes are available as modules. This module has a PHY already integrated (a Marvell Technology Group 88E1111 ultra gigabit Ethernet transceiver), which can handle 10/100/1000-Mbps Ethernet.

The TE0600 has a baseboard, so I didn't have to develop a board to get started. OpenCores's tri-mode MAC is published under the GNU Lesser General Public License (LGPL) and was created in 2005. The description language is Verilog; however, it is usually not a problem to mix Verilog and VHDL in one project.

While a User Datagram Protocol (UDP) core is available from OpenCores, I did not use the core in my project. I chose a UDP because it is much simpler to implement than a TCP/IP.

For example, you do not have to retransmit packets if they become corrupted or lost during transmission. Furthermore, the UDP is superior for streaming data whenever the data integrity is not so important or the latency has to be kept minimized.

WHY ETHERNET ON AN FPGA?

One goal of Ethernet on an FPGA may be to stream a lot of measured signals to a computer to analyze, display, or save. It can also be used to connect a camera over Camera Link to the FPGA and stream the captured images to a computer.

Another goal could be to stream signals

to the FPGA from a computer to have a signal generator with a lot of output channels.

There are complete solutions for these kinds of problems. They are available as National Instruments LabVIEW cards or DSpace software, but they are expensive and they are usually not customizable. Yet another advantage to using an FPGA is the ability to connect special hardware without any further overhead and excessive use of the parallel capabilities.

I want to show how to use OpenCores's open-source tri-mode MAC implementation and stream UDP packets with VHDL over Ethernet. It is even possible to use the free version of ISE with the LX45 module to implement UDP streaming.

I used VHDL as the hardware description language (HDL). P. J. Ashenden's *The Designer's Guide to VHDL* provides a good description of VHDL.

First, I will address the core's interface. Then I will explain the Ethernet format, the Address Resolution Protocol, and the UDP. Finally, I will present the implementation aspects and a UDP monitor example.

ETHERNET

Ethernet, a commonly used LAN technology, was introduced in 1980 and later standardized as IEEE 802.3. Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses (MAC addresses).

As per the Open Systems Interconnection (OSI) model, Ethernet provides services up to and including the data link layer. The OSI model includes: Layer 1, physical; Layer 2, data link; Layer 3, network; Layer 4, transport; Layer 5, session; Layer 6, presentation; and Layer 7, application.

Ethernet acts on Layer 1 similarly as Bluetooth or USB. Ethernet and the different protocols (e.g., TCP, UDP, etc.) are well addressed in C. M. Kozierok's *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. To communicate with IPv4 between two network devices, the MAC and both devices' IP addresses are needed.

THE ADDRESS RESOLUTION PROTOCOL

The ARP is used for the resolution of a network layer address (i.e., IP address into a link layer address or MAC address). ARP is a Layer 2 protocol because it is used to resolve a Layer 2 address. It is also a Layer 3 protocol because IP is a Layer 3 address.

Figure 1 shows the packet structure when using IPv4. The value of "Operation" specifies the operation the sender is performing. 0x01 is used for a request and 0x02 is used for a reply.

It is necessary to use ARP; otherwise the MAC address is unknown. So before sending UDP packets, an ARP request is sent to ask what MAC address has the IP address. The reply is taken and the MAC address is extracted. Because the destination's MAC address is unknown, 16'hFFFFFFFFFFFF is used, which represents a broadcast.

Furthermore, if a request with the FPGA board's IP address is received, you reply with a defined MAC address. It is also possible to use static ARP entries and program the destination MAC address directly in the source code. (Note: when using Linux, static ARP entries can be set like this: arp -s 192.168.1.1 00:1d:7e:xx:xx:xx). However, if more than one computer is used, that is not a practical approach. Usually, when the Ethernet card changes, the MAC address also changes.

0	7	15	23	31
Destination MAC address 1–2				
Destination MAC address 2–2		Source MAC address 1–2		
Source MAC address 2–2				
Ethernet type (0x0806)		Hardware type (0x0001)		
Protocol type (0x0800)		Hardware address length (0x06)	Protocol address length (0x04)	
Operation		Source MAC address 1–2		
Source MAC address 2–2				
Source IP address				
Destination MAC address 1–2				
Destination MAC address 2–2		Destination IP address 1–2		
Destination IP address 2–2		0x0000		

FIGURE 1
An IvP4 ARP packet structure is shown.

FIGURE 2
This is the IvP4 UDP's packet structure.

0	7	15	23	31
Destination MAC address 1–2				
Destination MAC address 2–2		Source MAC address 1–2		
Source MAC address 2–2				
Ethernet type		Version header	Different services	
Total length		Identification		
Flags/fragment offset		Time to live	Protocol	
Header checksum		Source IP address 1–2		
Source IP address 2–2		Source IP address 1–2		
Source MAC address 2–2		Source port		
Destination port		Length		
Checksum		Data		
More data content				

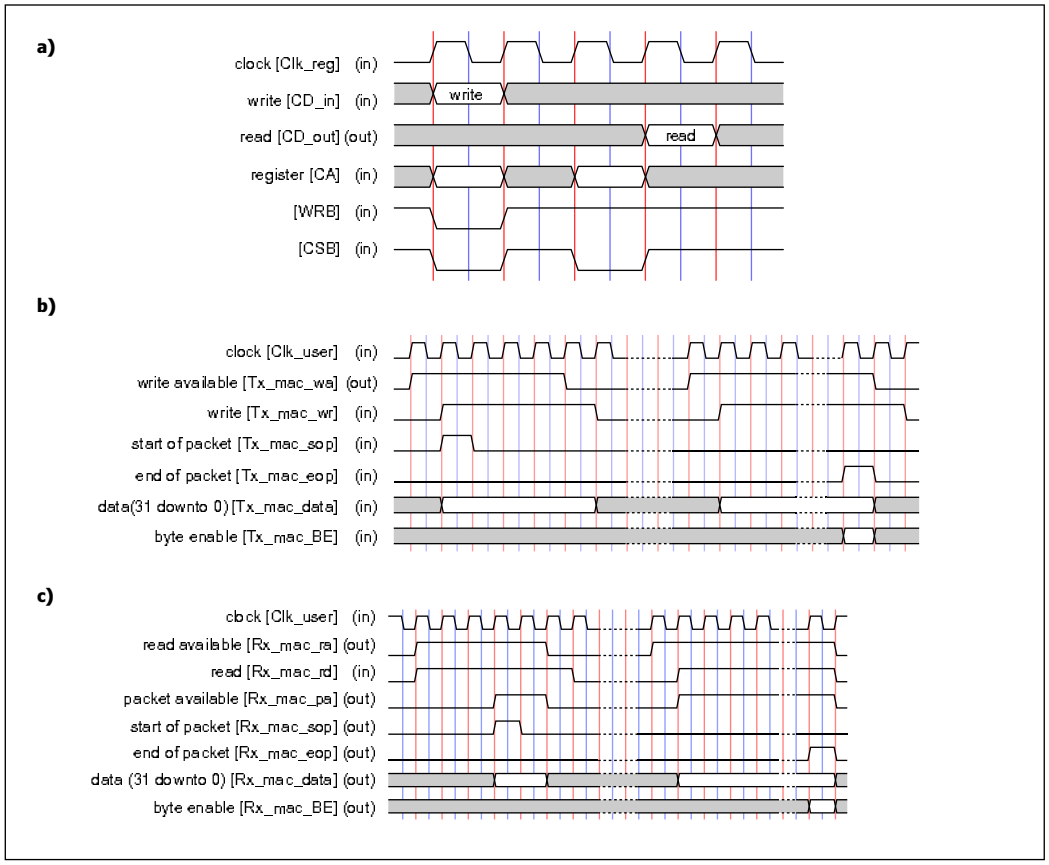


FIGURE 3
The core has one interface for transmission, one for receiving, and one for the host interface. **a—**This is the host interface's timing diagram. **b—**The read interface's timing diagram is shown. **c—**This is the write interface's timing diagram.

THE USER DATAGRAM PROTOCOL

Computer applications can use the UDP protocol to send messages (referred to as datagrams) to other IP hosts, mostly IPv4. It can also be used to network without prior communications to set up special transmission channels or data paths. In the OSI model, it acts on Layer 4, which is the transport layer.

UDP uses a simple transmission model with a minimum of protocol mechanism. Unlike TCP, it does not utilize handshaking. Therefore it is, by default, an unreliable protocol, which means that there is no guarantee of delivery, duplicate protection, or ordering.

UDP provides checksums for data integrity and port numbers to address different functions at the source and the datagram's destination. The checksum is optional. UDP is suitable when error checking and correction is either unnecessary or performed in the application. It avoids the overhead of processing at the network interface level and makes the hardware processing (e.g., FPGAs) a lot easier.

Time-sensitive applications (e.g., real-time data streaming) often use UDP because dropping packets is preferable to waiting for delayed packets. In a real-time system (e.g., video streaming), it is not a good option to wait for delayed packets.

Figure 2 shows the UDP packet structure when using IPv4. The checksum over the data is optional. If it is not used, it is normally set to 0x0000.

Note: When developing an Ethernet application or Ethernet capable devices, it is helpful to use a packet sniffer (e.g., the Wireshark free network protocol analyzer) to analyze the packets.

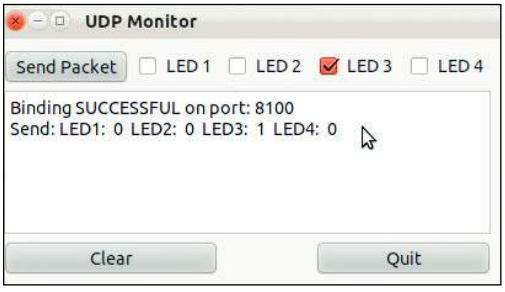


PHOTO 2
This screenshot shows the Qt4 UDP monitor.

OPENCORES'S TRI-MODE MAC

The 10-, 100-, and 1,000-Mbps tri-mode Ethernet MAC implements a MAC controller and is licensed under the LGPL, so it is open source. The implementation conforms to the IEEE 802.3 specification.

The MAC was designed to use less than 2,000 logic cells/logic elements to implement full function and is written in Verilog. A GUI

```
QUdpSocket *udpSocket = new QUdpSocket(this);
bool res;
int port = 8100;
res=udpSocket->bind(QHostAddress::Any,port);
QString temp;
QTextStream(&temp) << " Binding " << QString(res ?"SUCCESSFUL":"ERROR") << "on port:" << port;
// connects listener, when data is received processPendingDatagrams() is called
connect(udpSocket, SIGNAL(readyRead()), this, SLOT(processPendingDatagrams()));
```

LISTING 1
This code is used to create an UDP object in Qt4.

is available to configure optional modules; first in, first out (FIFO) depth; and verification parameters.

The core has three interfaces: one for transmission, one for receiving, and one for the host interface. With the host interface, for example, the Ethernet's speed can be set. Also, you can access statistics about the number of packets that have been sent (see **Figure 3a**).

The "read interface" reads the received data (see **Figure 3b**). When Rx_mac_ra is high, data in the FIFO is available.

Every signal is clocked in with the rising edge of Clk_user. Therefore Rx_mac_rd is set to high as long as Rx_mac_ra is high. When Rx_mac_pa is high, then Rx_mac_data is valid.

Rx_mac_sop and Rx_mac_eop signal the packet's start and end. Rx_mac_BE signals if the entire four bytes or only one, two, or three bytes are valid.

Figure 3c shows the "write interface." This interface is used to write data with the rising edge of Clk_user into the FIFO so the MAC core can transmit these to the PHY.

When Tx_mac_wa is high, there is space in the FIFO. Therefore Tx_ma_ca is set high and Tx_mac_sop is used to signal the packet's start.

Tx_mac_data is the data transmitted into the FIFO. If Tx_mac_wa is low, then

```
QByteArray datagram;
datagram.append(100);
QHostAddress myBroadcastAddress = QHostAddress
  ("192.168.1.1");
udpSocket.writeDatagram(datagram.data(), datagram.size(),
  myBroadcastAddress, 8100);
```

LISTING 2
This code is used to send a UDP packet in Qt4.

you have to pause until Tx_mac_wa is high again.

Tx_mac_eop is used to signal the end of the packet. Tx_mac_BE signals if the complete four bytes or only one, two, or three bytes have to be written.

The host interface is used to access the register to manipulate the core. When CSB is low and WRB is high, then a read operation occurs and CA specifies the register's address (see Resources for documentation about the tri-state MAC).

During the next rising edge of Clk_reg, the value is written to CD. If CSB and WRB are both low, a write operation occurs.

SPECIFIC DETAILS

It is necessary to use an ODDR2 on the Spartan-6 platform to drive the Ethernet TX CLK pin. Otherwise an error during implementation will occur, because a clock network should not drive an output pin.

Note: ChipScope is a powerful tool to use during debugging, but sometimes the signals are not available due to trimming. This can be prevented by defining the "save" attribute:

```
attribute save: string ;
attribute save of sig : signal is
      "TRUE"
```

The speed register, which is located at 7'd034, usually has a 16'h0004 default value. This register is used to set the Ethernet MAC core's speed level. 16'h0004 means 1,000 Mbps, 16'h0002 means 100 Mbps, and 16'h0001 means 10 Mbps.

The default value could be changed in reg_int.v. The speed register's value has to match the PHY's mode, which the PHY has negotiated (see the note below). If the FPGA board is directly connected to a PC, it is possible to limit a gigabit Ethernet card to 100 or 10 Mbps, for example.

Note: This implementation lacks the network speed's automatic detection, which the PHY negotiates with the switch, network card, or whatever. The automatic detection could be done by either using the 88E1111 PHY's status LEDs—which is a kind of hack—or using the MDIO interface to manually get and set the PHY's status (see Resources for more information).

OPERATION MODE

To send and receive UDP packets, you must know each receiver and sender's MAC and IP address or use the ARP to resolve them (see the Address Resolution Protocol section of this article).

In this implementation, the FPGA sends a broadcast ARP request with its own MAC and IP and the IP address of the destination and waits for an ARP response. When an ARP response from the destination IP address has been received, data transmitting and data receiving begins. Broadcast means

PROJECT FILES

circuitcellar.com/ccmaterials

RESOURCES

P. J. Ashenden, *The Designer's Guide to VHDL*, 3rd Edition (Systems on Silicon), Morgan Kaufmann, 2008.

J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt4*, 2nd Edition, Prentice Hall, 2008.

dSPACE GmbH, www.dspace.com.

GitHub, Inc., "UDP Network with a Tri-Mode MAC on a Spartan-6 FPGA."

C. M. Kozierok, *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*, 1st Edition, No Starch Press, 2005.

D. Molkentin, *The Book of Qt 4: The Art of Building Qt Applications*, 1st Edition, No Starch Press, 2007.

OpenCores, "10_100_1000 Mbps Tri-Mode Ethernet MAC: Overview."

———, "UDP/IP Core: Overview."

Trenz Electronic GmbH, "FPGA Boards."

Wikipedia, "Address Resolution Protocol."

———, "Management Data Input/Output."

———, "User Diagram Protocol."

Wireshark, free network protocol analyzer, www.wireshark.org

SOURCES

88E1111 Ultra gigabit Ethernet transceiver
Marvell Technology Group, Ltd. | www.marvell.com

LabVIEW Software
National Instruments Corp. | www.ni.com

TE0600 FPGA Micromodule
Trenz Electronic GmbH | www.trenz-electronic.com

Tri-Mode Ethernet Media Access Controller (TEMAC) core, AXI Ethernet Lite MAC, Embedded Development Kit (EDK), and Spartan-6 LX FPGA
Xilinx, Inc. | www.xilinx.com

ABOUT THE AUTHOR

Steffen Mauch (steffen.mauch@gmail.com) received his BSc in Electrical Engineering in 2010 and his MSc in Microsystem Engineering in 2011 from Furtwangen University, Baden-Württemberg, Germany. He is currently working as a research assistant in the Control Engineering Group at the Ilmenau University of Technology, Thuringia, Germany. Steffen's interests include robust control and real-time FPGA solutions.

that the MAC address is unspecified with 16'0hFFFFFFFFFFFF.

UDP MONITOR EXAMPLE

A Qt4 application is used to receive and send UDP packets. Two good references about starting to use Qt4 are J. Blanchette and M. Summerfield's *C++ GUI Programming with Qt 4* and D. Molkentin's *The Book of Qt 4: The Art of Building Qt Applications*.

The UDP port is defined as 8100 and the application also listens to this port. As shown in **Photo 2**, four LED checkboxes have been used to control the LEDs on the carrier board. If a checkbox is checked, this LED on the carrier board shines. UDP handling is simple when using Qt4.

Listing 1 shows how a `QUdpSocket` is created and how the `readyRead()` signal is used. As demonstrated, only a few lines of codes are needed.

Listing 2 shows how a datagram is sent. The datagram is created and an integer of 100 is added. Then the created datagram is sent to 192.168.1.1 on port 8100. These two listings are enough to receive and transmit data over Ethernet from the FPGA board.

DON'T FEAR AN FPGA

This article demonstrates that even a UDP implementation in an FPGA doesn't need to be intimidating. While using OpenCore's tri-mode MAC, the implementation is based on an open-source core, which enables its use without any fear of copyright infringement.

This design could be easily expanded. I kept the implementation simple without much error handling. If the transmitted data must always be correct, then some checksum and handshaking has to be implemented, but often it does not really matter.

The complete Xilinx ISE project and the Qt4 application with source files are available on the GitHub online project sharing community (see Resources). The complete code is published under the GNU LGPL.

FEATURES

Calibration (Part 2)

Polynomial Curve Fitting

The first part of this article series introduced linear calibration and recalibration of A/D conversion readings. It described how to convert A/D readings to real-world engineering unit values. The article also explained how to convert back from those values to the equivalent A/D readings. This article presents a method for fitting a polynomial curve to nonlinear data to perform nonlinear conversions. It also outlines an approach that converts back from the engineering units to the A/D readings.

By David Cass Tyler (USA)

Experimenting with a new project often means instrumenting it to gather data about what is going on. Part of that instrumentation process involves properly calibrating the inputs to get precise and accurate readings. By "precise" I mean taking measurements with sufficient resolution (decimal places) to see the tiny changes. "Accurate" means taking exact measurements. When an input reads 5 V, I want to know that the actual value is equal to or greater than 4.95 V and less than 5.05 V. I don't know for certain that second decimal place's value, but I do know the exact value rounded to the first decimal place.

"The greater the number of observed values across the measurement range, the better and more accurate your fit will be."

The first part of this article series explained how to calibrate and recalibrate linear measurements where real-world engineering units vary proportionally to the ADC values. Just like everything else, the real world does not always act that way. Not all measurements wind up being linearly proportional. So, how can you handle mysterious, unknown values?

You can take a series of measurements across the full range of inputs and produce a curve that fits them and a method to calculate the exact values. The greater the number of observed values across that measurement range, the better and more accurate your fit will be.

The best way to figure this out is to start by taking three readings at low, middle, and high points along the range. If they can all be connected by a straight line, they are a linear fit, right? Not necessarily.

What about an "S" curve? You can connect low, middle, and high points in an "S" curve with a straight line, but it won't describe the points that lie between the middle and the extremes. So, you take two more readings—one between the low and the middle points and one that lies between the middle and high points. If those points also fall on a straight line, chances are it is a linear calibration. This is actually how manufacturers gather data for their datasheets. They take enough data to see the underlying curve and then, knowing what the curve actually looks like, they take "pretty" values to get nice-looking curves for publication.

Once you have acquired enough points to accurately describe the underlying curve, you still want to be able to convert your A/D values into engineering units. This article explains how you can do that.

LINEAR INTERPOLATION

Linear interpolation is the simplest (and least accurate) way to convert your A/D values into engineering units. You gather an array of ordered pairs (A/D reading and the corresponding real-world engineering units) sorted by the X (A/D) value. Then, you look up the values that are immediately less than and immediately greater than the ones you want to convert. Then you can use the straight line formula to interpolate between them. Think of it as using short straight lines to describe the curve. The derived value can be fairly close if you fit enough line segments to the curve.

SPLINE INTERPOLATION

Another way to process this data is to use something like a cubic spline interpolation to fit a smooth line through all of the known points so you can calculate unknown values lying between them.[1] The method is similar to using a flexible ruler to connect the points.

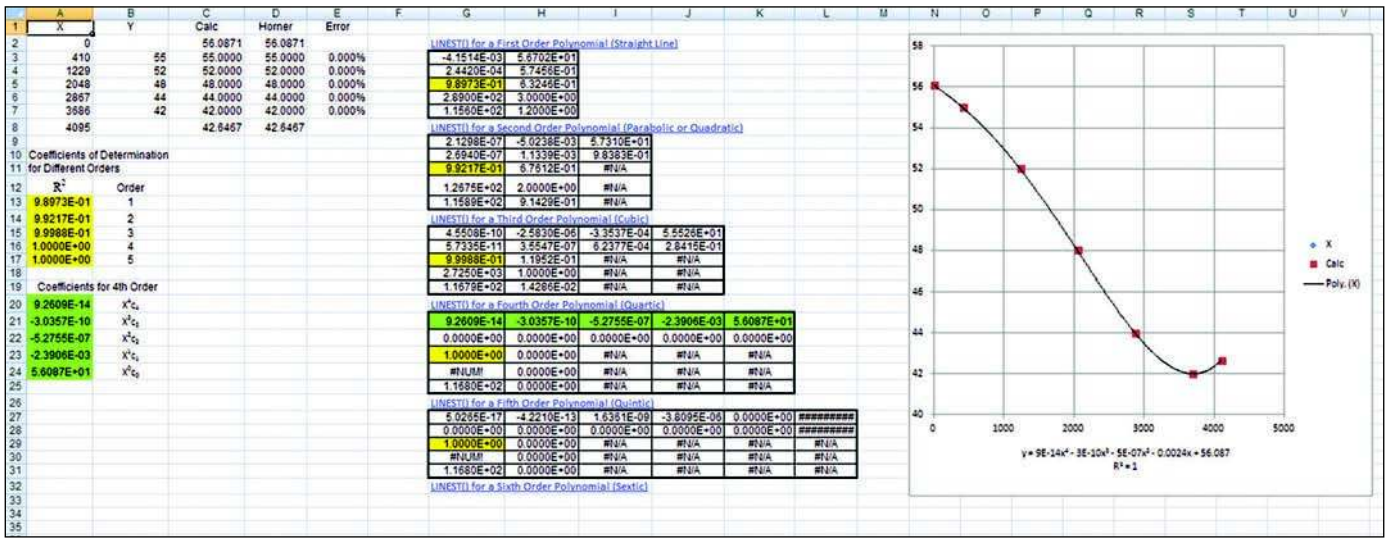


FIGURE 1
This spreadsheet displays the fourth-order chart values.

Books such as W. H. Press et al's *Numerical Recipes in C: The Art of Scientific Computing* provide the code necessary to use this method. The disadvantage of these methods is the amount of programming involved and the difficulty of understanding. There must be an easier way.

POLYNOMIAL CURVE FITTING USING EXCEL

One of the more commonly occurring curves is polynomial. A polynomial curve is expressed as:

$$Y = X^n c_n + X^{n-1} c_{n-1} + \ldots + X^2 c_2 + X^1 c_1 + X^0 c_0$$

In fact, a straight line equation is the most trivial polynomial where Y's value changes (an order 0 polynomial is just the constant c_0, since X^0 always equals 1). All you need to determine the engineering unit's value that is being measured is the A/D reading and an array of calibrated coefficients. There are many sources of information for curve fitting data to a polynomial. Again, *Numerical Recipes in C* is a good example.

For those of us looking for expedience, there is an easier way! You can use an Excel spreadsheet to calculate a polynomial that will enable you to easily compute the intervening values.

First, build an Excel two-column array where the first column holds the independent variable (the A/D reading) and the second column holds the associated dependent variable (the real-world engineering units). Take a bunch of measurements, starting at the low end of the range, but above the lowest A/D value, and work toward (but not to) the highest A/D value (i.e., stay off the rails).

Add blank values for the lowest and highest A/D readings so the chart encompasses the full range. To make it easier to work with, sort this array in ascending order. Select all of the values in both columns, including the column labels, and choose "Scatter" from the "Insert" tab. Choose "Scatter with only markers" as you will be "connecting the dots" with a trend line. Click on one of the graph's "dots." This will select all of them. Right click and select "Add Trendline …" to bring up the "Format Trendline" dialog box.

From this dialog box, you can draw different trend lines and choose one that best connects the dots. Check the box that says "Display R-squared value on chart." This is the "Coefficient of Determination" and it represents how well the trend line fits the data. The values range from 0 to 1, with 1 being the highest degree of fit.

The default trend line is the linear fit, which is a straight line's least-squares fit. Note its R2 value and then select polynomial and dial up the order from two to six, noting the corresponding R2 values and watching how well the trend line overlays the dots.

Choose the one that best fits the line as your calibration curve. Now click "Display equation on chart" to display the coefficients you will use to convert from your ADC reading to your real-world engineering units. Right click on the displayed equation, select "Format Trendline Label…" and select "Scientific" from the "Number Category." Set the label to display enough decimal places to represent the "Effective Number of Digits." This will give you your maximum usable precision during conversion.

Notice that the measurements you have are only two significant digits long. Even so, you have described a line that reflects the process. The calculated values will have a precision that is greater than two significant digits because the curve's shape dictates

that precision. Think of it as if you are trying to determine your speedometer reading's accuracy. As you go past a mile marker on the highway, you reset the trip odometer to zero. As you pass the tenth mile marker, you look down and see that your trip meter reads 10.1 miles. You now know your speedometer's precision to one decimal place. To determine the accuracy to two decimal places, repeat this process, but drive 100 miles and note the reading. For three decimal places, you would have to drive 1,000 miles. To increase your calculated readings' accuracy and precision, take more and more calibration points (evenly distributed and closer to the rails). They will tend to average out the errors.

"All you need to determine the engineering unit's value that is being measured is the A/D reading and an array of calibrated coefficients."

In **Figure 1**, the coefficients displayed in the equation are the numbers you are seeking. You can diagnose how well your coefficients work by calculating the Y values for the original X values and displaying them on the original chart. If the markers for the calculated values overlay the markers for the observed values, the curve is a fairly good fit. You can also compute the offset error and display it for further assurance of good coefficient values.

I highly recommend that you take this step. It will help flush out any problems and prevent them from cropping up later when they can become very expensive to correct.

You can use Horner's rule to effectively compute the Y values. Horner's rule states that the powers of X are factored out so the polynomial values are computed using a series of multiply/accumulates. The algorithm works for any polynomial order.

$$Y = X^3 \times c_3 + X^2 \times c_2 + X^1 \times c_1 + X^0 \times c_0$$
$$Y = X \times (X^2 \times c_3 + X \times c_2 + c_1) + c_0$$
$$Y = X \times [X \times (X \times c_3 + c_2) + c_1] + c_0$$
$$Y = X \times [X \times (X \times (c_3) + c_2) + c_1] + c0$$
$$Y = X \times \{X \times [X \times (0 + c_3) + c_2[+ c_1\} + c_0$$

Codified:

```
double X, Y, c[] = {c3, c2, c1, c0};
for (int i = 0, X = 0.0, Y = 0.0;
    I < n; i++) Y = Y*X + c[i];
```

This has been implemented in the `EvaluatePolynomial()` function in the included code samples (see Project Files) and will take the array of coefficients as a parameter. By factoring out the X values and putting the accumulation of factor values into a for loop, you eliminate the need to calculate the powers of X, making the computation much more efficient. Notice that it is now another multiply/accumulate operation.

In the spreadsheet, the C2:C8 range calculates the values using the powers of X while the D2:D8 range uses Horner's rule to calculate the same values. The formulas being used are:

```
=+A2^4*$F$21+A2^3*$G$21+A2^2*$H$21+
A2*$I$21+$J$21
```

and

```
= A2*(A2*(A2*(A2*($A$10)+$B$10)+$C$10)+
$D$10)+$E$10
```

When you recognize the Horner's rule pattern, it is actually easier to type it in for any given order.

The `LINEST()` function is used to calculate the equation displayed in **Figure 1**. This function returns an array. The five tables that are displayed are the `LINEST()` arrays for first- through fifth-order polynomials. Notice that column numbers vary according to the order. For instance, the table in the F21:J25 range is five rows and polynomial order plus one column. The table is specified as:

```
{=LINEST($B$3:$B$7,$A$3:$A$7^{1,2,3,
4},TRUE,TRUE)}
```

Select the entire range and go to "Formulas/Insert Function" to pop up the "Insert Function" dialog box. Select "LINEST" to get the "Function Arguments" dialog box. Select the ranges for "Known_ys" and "Known_xs" to specify the input data. Set "Const" to "TRUE" to compute the Y Intercept normally and set "Stats" to "TRUE" to output the regression statistics. This will set the upper left hand cell to:

```
=LINEST($B$3:$B$7,$A$3:$A$7,TRUE,TRUE)
```

To see the data for a fourth-order polynomial, edit the cell normally and add the "^{1,2,3,4}" to the "$A$3:$A$7" parameter. The first-order polynomial is specified as "^{1}." The second-order polynomial is specified as "^{1,2}." The third-order polynomial is specified as "^{1,2,3}." The fourth-order polynomial is specified as "^{1,2,3,4}." The fifth-order polynomial is specified as "^{1,2,3,4,5}." The sixth-order polynomial is specified as "^{1,2,3,4,5,6}."

The curly brackets specify that this is an array function. This is important! They must be added by selecting the data and pressing "CTRL+SHIFT+ENTER" to add the formula as an array. If you have problems, copy and delete the formula from the white formula

HuMANDATA LTD. www.hdl.co.jp/CC/ E-mail : s2@hdl.co.jp Fax : +81-72-620-2003

HUMANDATA® FPGA / CPLD Boards from JAPAN

SAVING COST=TIME with readily available FPGA boards

- Basic and simple features, single power supply operation
- Quality and reliability is provided by years of sales
- Same board size and connector layout – ACM/XCM series
- All stocked items are ready to be shipped immediately
- Over 100 varieties of FPGA/CPLD boards are available
- Customizing speed grade and/or any features are possible
- Free download technical documents before purchasing
- High quality and highly reliable FPGA/CPLD boards from Japan
- Almost all products are RoHS compliance

ALTERA FPGA Board

Cyclone IV E F780 FPGA board

ACM-204 series NEW

Cyclone IV E | SDRAM | SW | LED | I/O:296

EP4CE30F29C8N
EP4CE40F29C8N
EP4CE115F29C8N
Credit card size (86 x 54 mm)
RoHS compliant

Cyclone III F780 FPGA board

ACM-202 series

Cyclone III | SW | LED | I/O:296

EP3C55F780C8N
EP3C80F780C8N
EP3C120F780C8N
Credit card size (86 x 54 mm)
RoHS compliant

Cyclone IV GX F484 FPGA board

ACM-108 series

Cyclone IV GX | DDR2 | SW | LED | I/O:128

EP4CGX50CF23C8N
EP4CGX110CF23C8N
EP4CGX150CF23C7N
Compact size (43 x 54 mm)
RoHS compliant

MAX II T144 CPLD board

ACM-302-1270

MAX II | LED | SW | I/O:56

EPM1270T144C5N
Compact size (54 x 53 mm)
RoHS compliant

USB- FPGA Board

Cyclone IV USB-FPGA Board

EDA-301 NEW

Cyclone IV E | USB Config. | USB Comm. | HI-SPEED

EP4CE15F17C8N
Compact size (54 x 53 mm)
RoHS compliant

XILINX FPGA Board

Spartan-6 FGG484 FPGA board

XCM-019 series

Spartan-6 | 5V I/O | SW | LED | I/O:100

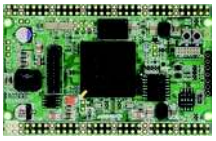
XC6SLX45-2FGG484C
XC6SLX75-2FGG484C
Credit card size (86 x 54 mm)
RoHS compliant

Spartan-6 LXT FGG484 FPGA board

XCM-020 series

Spartan-6 | MRAM | DDR2 | RocketIO | SW | LED | I/O:100 | SIF40

XC6SLX45T-2FGG484C
XC6SLX75T-2FGG484C
XC6SLX100T-2FGG484C
XC6SLX150T-2FGG484C
Credit card size (86 x 54 mm)
RoHS compliant



Virtex-5 FFG676 FPGA board

XCM-109 series

Virtex-5 | SDRAM | SW | LED | I/O:128

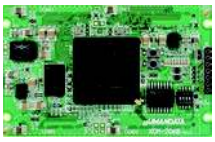
XC5VLX30-1FFG676C
XC5VLX50-1FFG676C
XC5VLX85-1FFG676C
XC5VLX110-1FFG676C
Compact size (43 x 54 mm)
RoHS compliant

Spartan-6 FGG676 FPGA board

XCM-206 series

Spartan-6 | MRAM | DDR2 | LED | SW | I/O:296

XC6SLX100-2FGG676C
XC6SLX150-2FGG676C
Credit card size (86 x 54 mm)
RoHS compliant



Spartan-6 USB-FPGA board

EDX-301 NEW

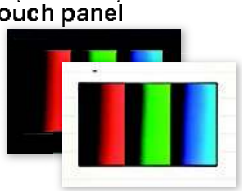
Spartan-6 | USB Config. | USB Comm. | HI-SPEED

XC6SLX16-2CSG225C
Compact size (54 x 53 mm)
RoHS compliant

5" LCD Touch Panel Module

5 inch TFT full color LCD display with WVGA(800x480) resolution resistive touch panel

NEW **UTL-021**



- 3.3 V single power supply operation
- Piezo buzzer to beep
- Useful plastic bezel is included to assemble
- LTM-compatible pin assignment

FPGA/CPLD Stamp Module PLCC68 Series

Easy and Quickly Mountable Module

FPGA Module IC socket mountable

- 50 I/Os (External clock inputs are available)
- 3.3V single power supply operation (Voltage converters for auxiliary power supply are built-in)
- Separated supply-inputs: Core, I/O drivers
- JTAG signal
- All PLCC68 series have common pin assignment
- Very small size (25.3 x 25.3 [mm])
- RoHS compliance
- MADE IN JAPAN

XILINX PLCC68 Series

Spartan-6 PLCC68 FPGA Module

XP68-03

Spartan-6 | PLCC 68

XC6SLX45-2CSG324C
3.3V single power supply operation
On-board oscillator, 50MHz
RoHS compliant

ALTERA PLCC68 Series

Cyclone III PLCC68 FPGA Module

AP68-04

NEW Cyclone III | PLCC 68

EP3C25U256C8N
3.3V single power supply operation
On-board oscillator, 50MHz
RoHS compliant

Universal Board (Type2)

ZKB-106

- One for general power(3.3V 3A max) and the Two variable outputs for Vccio(0.8V to 3.3, 3A max)
- For ACM/XCM-2 series FPGA boards
- Power Switch and LED
- Power Input: DC5V/2.1[mm] Jack/ Terminal Block (option)
- Board size : 156x184 [mm]
- 4 Layers PCB, Thru-hole

See all our products, A/D D/A conversion board, boards with USB chip from FTDI and accessories at: **www.hdl.co.jp/CC/**

bar at the top of the page. Select the 5 × 5 cell range.

All the cells will be blue except the top left cell, which will be white. Use the mouse to select the formula bar at the top and paste the LINEST formula back into the line. Press the "CTRL+SHIFT+ENTER" to add it as an array. Now select the output range and format the numbers as "Scientific" to four decimal places.

You don't have to display the entire table. You can display only part of it by using the `INDEX()` function as shown in the A13:A17 range. These are the "Coefficients of Determination" for the first five orders. They are what is displayed on the chart as the "R2" number. They can enable you to determine the best order without going through all the graphing and so forth. The fourth-order polynomial's coefficients are displayed in the A20:A24 range in the same way.

The last thing I want you to notice is the charted point for "X=4095." See how it kicks back up at the end? Even though the calculated polynomial passes exactly through each and every point, it is *not* necessarily a good fit. You have to use enough input points to describe the curve for the entire range in which you are interested. You may have to use a different polynomial order and suffer the resulting loss of accuracy to get the most usable fit.

Always remember all these polynomials are just best-guess estimates, even if they are backed up with sound mathematics. You, the engineer, have to make the final well-reasoned decision.

DIGITAL TO ANALOG

There is always the problem of converting back from your Y value to the corresponding X value. I am including code to accomplish this as long as Y always increases (or decreases) as X increases. Please note that for polynomials that are two or greater, multiple X values can evaluate the same Y value. I am going to specify that for any given Y, there is a unique X value. You are on your own to answer this question in other circumstances.

At the end of my sophomore year of college, I ran out of money and had to drop out of school and get a job. I wound up spending eight years as a carpenter and cabinet maker, which you would think wouldn't relate very well to computers. You would be wrong.

Lessons in each field can be applied to other fields with good results. For instance, I always needed unknown-sized shims and I couldn't carry around everything I needed. I realized that for anything from 0.03125" to a full inch that I could just carry the powers of two fractional inch shims (i.e. 0.5", 0.25", 0.125", 0.0625", and 0.03125") and create anything I needed by adding them together. If I needed a 0.75" shim, I would use a 0.5" and a 0.25" shim to create it. Likewise, 0.875" would be 0.5" + 0.25" + 0.125", and so on.

Later, after I had read Freescale Semiconductor's "MCF5282 and MCF5216 ColdFire Microcontroller User's Manual," I realized that a successive approximation ADC used this same principle. In Chapter 28, "Queued Analog-to-Digital Converter (QADC)," the manual explains how the user supplies the high- and low-reference voltages, VRH and VRL, and how the successive approximation register sequentially receives the conversion value one bit at a time, starting with the most-significant bit. The Successive Approximation Analog to Digital SAAD() "C" function simulates this operation.[2]

Since I specified that Y always increases as X increases, I can ensure I have accumulated the correct ADC value. If you want to know the ADC equivalent to an analog value, you can use this function to find it.

I included a program to exhaustively evaluate each value the ADC can produce to ensure that `SAAD()` finds the correct value for each one (see Project Files). It does as long as all values of Y ascend or descend, as the case must be. The code sample uses the coefficients for the fourth-order equation as its test case.

This is an instance where the Y values decrease as X increases and it fails when testing for the 3585 value. This is because of the reversal of direction where Y starts increasing as X increases, which makes the value comparison fail. The only way you would know would be to run the test program with your set of coefficients (remember to set the number of bits) and see that all values pass.

I have also included the code to compute the binomial least-squares fit coefficients. It is

PROJECT FILES

circuitcellar.com/ccmaterials

REFERENCES

[1] *Wikipedia*, "Spline interpolation."

[2] Freescale Semiconductor, Inc., "MCF5282 and MCF5216 ColdFire Microcontroller User's Manual," 2009.

RESOURCES

I. Miller and J. E. Freund, *Probability and Statistics for Engineers*, 3rd Edition, Prentice Hall, 1985.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge University Press, 2002.

ABOUT THE AUTHOR

David Cass Tyler (David.Cass.Tyler@gmail.com) is a retired embedded systems engineer who lives in Willard, NM. He is the owner and sole author of *The Control Freak* (www.the-control-freak.com), which he uses to share his knowledge with the community. David is currently working on a Standard Commands for Programmable Instruments (SCPI) parser that he hopes to sell on his website.

implemented from I. Miller and J. E. Freund's, *Probability and Statistics for Engineers*, so you have the code to programmatically determine a second-order polynomial's coefficients.

The code matches what is computed in Excel in the spreadsheet (see Project Files). Functions have also been supplied to return the values of "Not a Number," "Positive Infinity," and "Negative Infinity" so you can return those values when appropriate.

METHODS FOR SUCCESS

The first part of this article series discussed how to calibrate and recalibrate linear conversions. This article examined the tools to handle up to sixth-order polynomials. The code for binomial curve fitting is available (see Project Files) and a discussion for using scatter charts and the `LINEST()` function in Excel, complete with examples, has been presented.

Using these polynomials enables you to expand the range of transducers you can choose for your projects. It also enables you to design and implement equipment that doesn't have to be linear.

Excel is the poor man's tool for doing this. Various curve-fitting programs do a nice job, if you can afford them. There are books that discuss curve-fitting algorithms, if you want to explore them. The methods presented in this article series are enough to get you started.

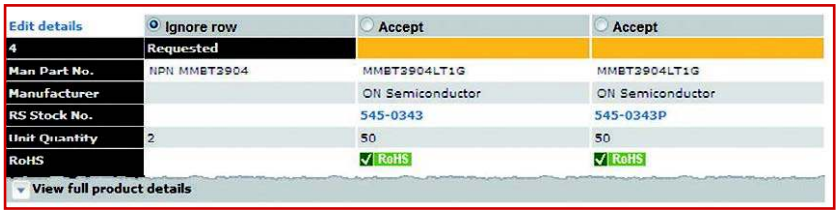
With these methods, you should be able to handle the majority of situations you are likely to encounter and achieve precise and accurate readings within your system. Good luck with your experiments and excursions into the unknown.

DesignSpark Tips & Tricks

Day #6: After the layout

By **Neil Gruending**
(Canada)

Today we're going to use the DesignSpark online BOM and PCB quoting tools to find out how much it would cost to build our example project. These tools can be a great time saver.



Edit details	Ignore row	Accept	Accept
4	Requested		
Man Part No.	NPN MMBT2904	MMBT3904LT1G	MMBT3904LT1G
Manufacturer		ON Semiconductor	ON Semiconductor
RS Stock No.		545-0343	545-0343P
Unit Quantity	2	50	50
RoHS		✓ RoHS	✓ RoHS
View full product details			

Figure 1.
Best matching of the generic type code 'NPN MMBT3904' to RS Components part numbers.

BOM Quotation

In our last installment we generated a bill of materials (BOM) as a spreadsheet that could be used to manually order the parts we need for the board. Some supplier websites will let you upload your BOM to order your parts, but DesignSpark cuts out the intermediate steps and connects to the RS Components website for you. The website doesn't work worldwide yet, but I hear that they're working on it. For this article, I set my locale to the United Kingdom (Settings → Preferences menu) since the website connection doesn't work for Canadians yet.

So let's see what will happens when we click on the BOM Quote, since the components I created didn't have any RS part numbers. Once you click on the button, DesignSpark will run the built-in BOM report using the fields that it knows about: Reference Designator, Quantity, Component Name, Component Value, Package Name, Manufacturer, Manufacturer Part Number, RS Part Number, and the Component Description. Note that the Package Name field is a separate field in the schematic symbol and not the PCB footprint name. Next, DesignSpark will log you into the RS website with your ModelSource ID so that the BOM can be uploaded to the RS website. The website will then do its best to match the component fields in the BOM to RS part numbers. When the RS part number field is blank, the website will propose its best matches.

For example, **Figure 1** shows the proposed matches for the MMBT3904 in our design. There you can see that the website is using the Component Name "NPN MMBT3904" as the main search term and that the website is proposing its closest matches. If you click on the View full product details arrow on the bottom row, the table will expand to show more details like the component cost. In this case, we'll accept the first match because it's the correct part number. **Figure 2** shows another example.

Edit details	Ignore row	Accept	Accept
2	Requested		
Man Part No.	0805 100 5%	RS-0805-3k6-5%-0.125W	RS-0805-3k9-5%-0.125W
Manufacturer		Viking Tech Corpor...	Viking Tech Corpor...
RS Stock No.		713-6982	713-6985
Unit Quantity	1	1	1
RoHS		✓ RoHS	✓ RoHS
Description	Resistor	0805 Resistor, 0.1...	0805 Resistor, 0.1...
Price		£8.63	£8.63
Units		1 Reel of 5000	1 Reel of 5000
Goods Price		£8.63	£8.63
Availability		In stock for next w...	In stock for next w...

Figure 2.
Here we find the RS Components part numbers for 3k6 (3.6 kΩ) and 3k9 (3.9 kΩ) SMD resistors.

This time the website couldn't find an appropriate match, which leaves a couple of options to find the correct part. If you have a RS account, then you can click on the Edit details link to modify the part information, but you will need to do this every time you upload a BOM that uses this part. For that reason I like to correct the part information in DesignSpark's libraries instead. The manufacturer part number "0805 100 5%" that was used by the website was actually the Component Name for R3 which means we should rename it. First, open the Library Manager and then navigate to the 100R component in the resistor library. Here you will find a Rename button so that you can rename the part "0805 100r 5%."

The next step is to tell DesignSpark to reload the part parameters for R3 from the library. Normally you would use the "Update Components → All Components" command in the Tools menu or right clicking on R3 and selecting "Update Component." But since we've changed the Component Name we'll need to replace R3 with the updated component by going into the Component Properties and clicking the Change button, see **Figure 3**. The "Change Component" window will then open so that you will be able to select our new resistor from the library. Now when you click the "BOM Quote" button the RS website will recognize the modified 100R resistor.

Some parts like the LED will be difficult to match with just the component name so you could set the RS part number in the "RS Part Number" field instead. Then just update the component instead of changing the component. Once all the components have been updated then you can create an order pad by pressing "Add accepted items to order pad" where you can see the total costs and place an order.

PCB Quotation

We'll also need a printed circuit board if we're going to make our design, so let's try using DesignSpark's PCB quoting function. As part of the quoting process, DesignSpark will check if you've run a design rule check (DRC) on the board and that it's within the design limits for the service. I had problems using Chrome as my default browser when trying the quotation function, so I used Internet Explorer for the rest of the quotation process. If you try and quote the circuit board as is, DesignSpark will warn you that the board is too small because the minimum size for quoting is 30 mm × 30 mm and our board is 20 mm × 20 mm. If you ignore the error and try and get a quotation anyway, the quotation website will fail. We will have to panelize our board to make it large enough to meet the minimum PCB size requirement.

Panelizing a circuit board refers putting multiple copies of the circuit board onto a larger PCB panel for manufacturing. This is how a circuit board manufacturer produces circuit boards, but there is a point where it becomes cost prohibitive to cut out smaller boards out of large panels. So what we will do for our board is to duplicate the board four times to make it 43 mm × 43 mm instead, as a 2 × 2 array of boards with 3 mm between them. The extra 3 mm gives you room to cut out the boards.

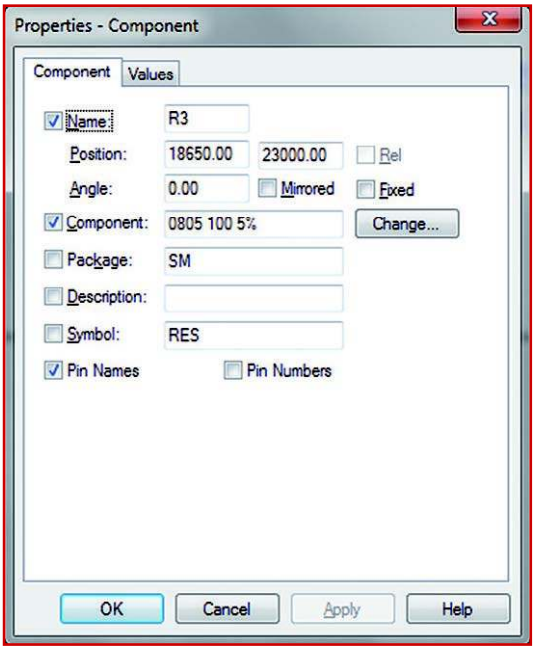


Figure 3.
The component renaming window.

DesignSpark can't automatically panelize our board for us, but it is possible to do it manually by selecting the entire board, copying it and then pasting it back into the PCB file. Just make sure that you choose to not to merge the 5 V and GND nets when DesignSpark asks. Unfortunately, DesignSpark will also automatically increment all of the reference designators so that they don't conflict with the rest of the boards. The only way to fix this is to manually edit, but this is tricky with DesignSpark because it requires that every component have a unique reference designator. One method is to give each board a unique suffix to the designators when renaming them. For example R1 could be R1A, R1B, R1C and R1D. DesignSpark will now be able to quote the panelized board, but you will still need to contract the PCB manufacturer to make sure that they are able to accept panelized boards.

Conclusion

Today we modified our design so that DesignSpark could give us BOM and PCB cost estimates. Next time we'll look at how DesignSpark can render a 3-D image of the board.

(130247)

FEATURES

Emerging Memory Technologies

A variety of memory technologies is available for use with embedded systems. This article examines some newer technologies and provides details about phase-change, magnetoresistive, ferroelectric, and resistive memory types.

By Faiz Rahman (USA)

PHOTO 1
Micron Technology offers phase-change RAM IC packages. (Image courtesy of Micron Technology)

Memories are pervasive in digital systems. Open up almost any modern computing, communication, entertainment, surveillance, or control system and you will find a handful of memory chips inside.

Digital data has to be stored in a suitable medium and made available as information processing devices work with it. Any digital data processor—a microprocessor, a microcontroller, a DSP, or a graphic processing unit (GPU)—must be paired with a suitable solid-state memory system to create a practical usable device.

Several different IC memory technologies are available for digital system designers. Some are better in certain performance specifications than others. No currently available technology can claim to excel in all respects. This disparity has sustained several different memory technologies, such as dynamic RAM (DRAM) and static RAM (SRAM), which feature high access speed, and flash memory, which is known for its nonvolatility and high integration density. Digital systems often comprise several different memory types to take advantage of their unique performance and economic benefits.

This article discusses some interesting new memory technologies that have become available for use in embedded systems. I cover only those devices that are now commercially available, but bear in mind that many other technologies are being hotly pursued in academic and corporate research labs worldwide. As I examine these emerging memory technologies, it is important to understand that it has taken many years, the sustained efforts of many individuals, and enormous expenditures by technology companies to bring each of these memory technologies to market. Thus, each memory technology described is a testament to human ingenuity.

PHASE-CHANGE RAM

One of the most interesting memory types to emerge in recent years is one that stores data as order or disorder in small islands of a special material. The structural transition between ordered and disordered phases is driven by controlled heating of the material island. Phase-change RAM (PRAM) has been in intensive development since the 1990s, but commercial products have only now become available.

It is well known that conducting materials, where atoms are regularly arranged, have high electrical conductivities compared to disordered conductors. This difference in conductivity is utilized by PRAM to store digital data. These devices use a "phase-change material," such as germanium antimony telluride ($Ge_2Sb_2Te_5$), which is abbreviated as GST. What is striking about this semiconductor material is that it can exist in both crystalline (ordered) and amorphous (disordered) forms, which can be transformed into each other by applying a very brief heat pulse.

GST's crystalline phase has orders of

magnitude lower resistivity compared to its amorphous phase. PRAM devices use GST's crystalline state to represent bit 1 and its amorphous state to represent bit 0. The two structural phases are stable, so phase-change memories are nonvolatile with data retention times in excess of 10 years at temperatures in excess of 100°C. With repeated read-write cycles or endurance of more than a million cycles, PRAM is the leading candidate to replace ubiquitous flash memory for mass-storage applications.

The main drawback of widely used flash memory is its slow access time. This makes it only suitable for use as a storage device. Flash memory also requires erasing an entire block before even 1 bit can be changed—a process that requires several milliseconds. PRAM, on the other hand, features write times of around 100 ns and read times of around 20 ns, making it much faster.

Figure 1 shows a typical PRAM cell's structure. The storage node consists of a tiny island of GST on top of a titanium nitride (TiN) heater. The GST island is sandwiched between top and bottom electrodes and is contained within a silicon dioxide dielectric envelope. A memory cell selector transistor is fabricated on top of the storage node. This cell access transistor serves to select an individual addressed cell to write and read data.

To perform a write operation, the transistor drives a heating current into the TiN heater. Writing a 1 bit requires a low-value current pulse for several tens of nanoseconds. This causes the phase-change material to anneal itself (i.e., to adopt a low-resistivity ordered crystalline state). On the other hand, to write a 0 bit, a short-duration high-current pulse is injected, which melts the GST material and cools it down quickly before its atoms can arrange themselves into an ordered structure. The resulting glassy amorphous phase has a high resistivity.

To read a PRAM cell's state, its access transistor is made to inject a small constant current through the GST island. The resulting voltage drop across the cell is measured to reveal whether the cell is in a low-resistivity ordered state (bit 1) or a high-resistivity disordered state (bit 0).

There have been several recent advances in PRAM technology. Perhaps the most remarkable is the ability to control the cell-heating current precisely enough to create several intermediate cell-resistance values. This immediately increases the memory capacity as each cell can be made to store more than one bit. For example, if eight resistance values can be created and distinguished, then the cell can be used to store three bits, thus tripling the memory capacity. This is now a routinely used technique implemented with PRAM devices.

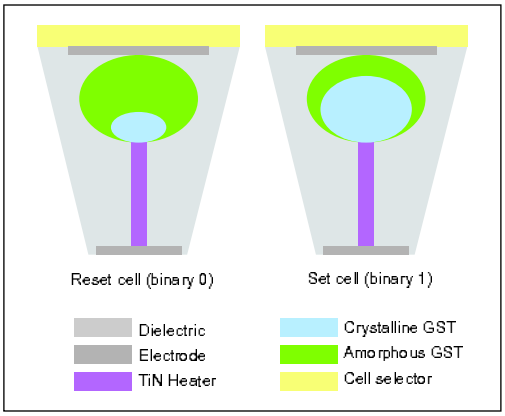


FIGURE 1
The structure of phase-change RAM cells in reset and set states is shown.

Another development, still in the research labs, is to stack several PRAM cell arrays on top of each other to create high-capacity chips. This fabrication technology has been demonstrated in principle but has not been implemented in real chips.

PRAM technology has historically been pursued by several companies, but there has been much consolidation over the years. Micron Technology and Samsung are currently leading PRAM chip development. Micron offers several PRAM chips in BGA and SOIC packages with capacities of 32, 64, and 128 Mb with operating speeds as high as 66 MHz. Both SPI and parallel interfaces are available, with the former targeted at flash-replacement applications and the latter intended for all other applications. **Photo 1** shows Micron's PRAM IC packages.

Samsung has its own aggressive marketing plan for PRAM devices and has demonstrated experimental chips with capacities as high as 512 Mb. The company will use its chips for in-house products and for sale to other OEMs.

MAGNETORESISTIVE RAM

Computer boot times are a constant source of annoyance to many computer users. As previously mentioned, the source of the problem is the necessity to read data

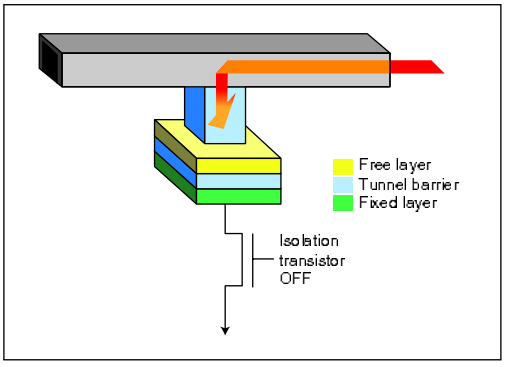


FIGURE 2
A spin-torque magnetoresistive RAM cell's structure includes a free layer, a tunnel barrier, and a fixed layer.

PHOTO 2
Everspin Technologies's EMD3D064M is a 64-Mb spin-torque magnetoresistive RAM IC. (Image courtesy of Everspin Technologies)

from a large, slow, but inexpensive memory (hard drive) to a small, fast, but expensive memory (DRAM), which the computer processor can directly access. We have all wished for a computer with no start-up delay that could be ready to use almost as soon as it was powered up. Such a computer will need to use an inexpensive but fast nonvolatile memory. This combination is difficult to come by, but proponents of magnetoresistive RAM (MRAM) think boot times could soon become outdated as this new memory becomes a mature product.

MRAM technology is based on using the direction of tiny magnetized regions as the means of encoding binary digits 0 and 1. In this respect, it is similar to legacy memory technologies (e.g., bubble memory and magnetic-core memory), which once dominated storage in mainframe computers.

MRAM contains an array of memory cells, each coupled to word and bit lines with an access transistor. Each MRAM chip's memory cell is made of a tiny sandwich of magnetic and insulating thin films called a magnetic tunnel junction (MTJ). The MTJ behaves as a two-valued resistor. The resistance is dependent on one of two possible magnetic states. This bistability enables an MTJ to store a 0 or 1, just as the presence or absence of charge enables a DRAM to store information. However, an MTJ's magnetic polarization states don't need any power to survive (exactly as any ordinary magnet retains its magnetism without any external energy expenditure). So, unlike DRAM and SRAM, memory based on MTJs requires neither periodic refreshing nor power application to retain its contents.

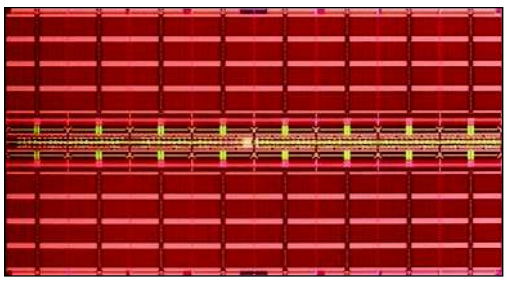


PHOTO 3
A 64-Mb spin-torque magnetoresistive RAM die from Everspin Technologies is shown. (Image courtesy of Everspin Technologies)

The tunnel junction is made of two layers of magnetic material separated by a thin layer of a material that is both nonmagnetic and nonconducting. One magnetic layer is made from a magnetically hard material and has a fixed permanent magnetic orientation. The other layer is made from a magnetically soft material and is capable of switching its polarity from one orientation to the opposite one. This switch can be affected in a few nanoseconds by a current flowing into the magnetically soft layer whose magnetic orientation is free to rotate.

The cell can be read by passing a current vertically through the MTJ. This current "tunnels" through the insulating layer between the two ferromagnetic layers. If the two ferromagnetic layers have the same magnetic orientation, then the cell has a low "magnetoresistance" and a large current flow. On the other hand, if the two layers have antiparallel orientation, then the cell's magnetoresistance is high and a smaller current flows through it.

Figure 2 shows the structure for a "spin-torque MRAM." By measuring the current flow, it is possible to tell the two magnetic orientation states apart and thus the bit stored in the MRAM cell. Laboratory tests have consistently shown that MTJs' magnetic states are very stable and unaffected by mechanical, thermal, and magnetic shocks. The rest of the circuitry needed to make a complete memory chip is essentially the same as with other CMOS solid-state memory types. Thus, MRAMs present the same external interface and communication protocols as any other semiconductor memory chip. This makes it easy to upgrade existing systems to use MRAMs instead of other conventional memories.

MRAM's nonvolatility alone will not make it a potential game-changing technology. Its high-access speed is what makes it special. Unlike other nonvolatile memory (e.g., EEPROMs and flash), MRAM boasts typical access speeds of 35 ns and potentially as short as 4 ns, with further developments. This combined with MRAM's extremely high endurance and data retention periods of more than 20 years even makes the technology suitable for use as CPU cache memories, which is a very demanding application.

One further advantage of MRAM is that its basic architecture—where the access transistor can be formed directly on top of the MTJ—enables very dense integration, greatly reducing the cost of storage per bit and making MRAM well suited for use in solid-state disks. Throw in its relatively low standby power dissipation of less than half a milliwatt, and MRAM also becomes attractive

for hand-held and portable devices. It is not difficult to see why this technology has been described as the "universal" memory device.

EMD3D064M MRAM is now available from Everspin Technologies (a spin-off from Freescale Semiconductor) in small-footprint BGA and DFN packages with either serial (SPI) or parallel interface (see **Photo 2**). Operating at a 3.3-V industry standard, these ICs are currently available with densities of up to 16 Mb per chip. **Photo 3** shows an Everspin Technologies 64-Mb MRAM die.

Other future vendors include Toshiba and Hynix Semiconductor, which have announced a joint venture for MRAM chip R&D and manufacturing. Some companies are already shipping products with small amounts of embedded MRAM. Buffalo Technology, for instance, is selling flash-based solid-state drives with an 8-MB MRAM cache memory. At the time of this writing, the high cost of MRAM devices and their relatively low capacities compared to that of other memory technologies remain as barriers to more widespread use. However, with ongoing developments, it is certain that in the near future, MRAM chip costs will reduce significantly while their storage capacities will rise, greatly increasing their usage in many digital systems.

FERROELECTRIC RAM

DRAM is at the heart of all contemporary computers due to its fast access times, high integration density, and low cost. We are all familiar with the "boot time" that occurs every time a desktop or laptop computer is switched on. During this time, the OS and other start-up programs are read from the hard drive and stored into DRAM banks where these can be executed by the microprocessor in real time.

In many ways, DRAM is an example of an ideal memory, if it weren't for its volatility. DRAM stores binary digits 0 and 1 as the absence or presence of electrical charge, respectively, on tiny capacitors—one per memory cell. Each capacitor is accessed by a dedicated access transistor for read and write operations.

The problem is that the charge stored in a DRAM cell tends to disappear due to self discharge after only a few milliseconds. This means that all DRAM chips have to be periodically read and every cell's state must be restored every few milliseconds. The requirement for periodic "refresh" operations increases the power consumption of DRAM banks, in addition to endangering data integrity in the case of even short power supply dips.

Within this backdrop, ferroelectric RAM (FRAM) became a potential game changer when it was introduced in the early 1990s.

FRAM is similar to DRAM but uses a different material for its capacitor dielectric. Therein lies the main difference between the two memory types. FRAM utilizes "ferroelectric materials" as memory cell capacitor dielectric material. A ferroelectric material—placed inside an electric field (e.g., a capacitor)—becomes polarized with a permanent electric field, which it retains even when the external polarizing field is removed. This essentially means that once a ferroelectric capacitor is charged and the voltage across it is removed, an electric field still remains between the capacitor plates. The direction of the polarization field depends on which way the capacitor was charged to

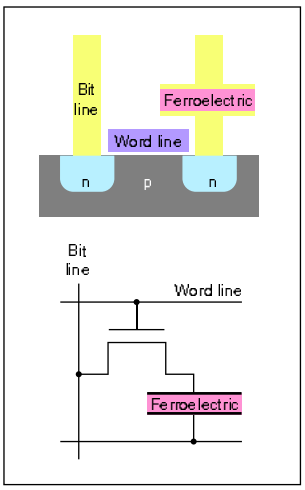


FIGURE 3
A ferroelectric RAM cell's organizational structure is shown.

PHOTO 4
This is Cypress Semiconductor's 1-MB parallel access FM28V100 ferroelectric RAM IC. (Image courtesy of Cypress Semiconductor)

	Access Speed (Read/Write Time)	Symmetric Access	Power Consumption	Endurance	Data Retention	Integration Density	Noise Immunity
PRAM	Medium	No	High	High	High	Medium	High
MRAM	Fast	Yes	Medium	High	High	High	High
FRAM	Fast	Yes	Low	High	High	High	High
ReRAM	Fast	Yes	Low	High	High	High	High

TABLE 1
This table compares the performance attributes of different emerging memory technologies.

polarize the ferroelectric dielectric.

This behavior is similar to ferromagnetic materials placed inside magnetic fields when they can be polarized to become permanent magnets. This similarity is why the term "ferro" exists in the word ferroelectric, otherwise these two classes of materials have little in common. A class of materials called "perovskite" has often been used in ferroelectric memory cell applications.

The permanence of induced electrical polarization in ferroelectric capacitors endows FRAMs with their nonvolatility. To write a particular bit, a FRAM's cell capacitor is briefly charged in one direction to polarize the ferroelectric material between its plates. The capacitor voltage can then be removed and the bit state will be retained in the directional sense of the dielectric material's polarization. No charges may leak away, and the polarization can be maintained for many years making FRAM, in a sense, a nonvolatile analog of DRAM.

Figure 3 shows a FRAM cell's organizational structure. The ferroelectric capacitor is connected to one of the cell selector transistor's terminals; the other terminal is the bit line. The transistor's gate is connected to the word line. When both the bit and word lines are driven high, the cell is selected and the desired memory operation is performed, depending on the potential at the other end of the capacitor.

Several companies have active FRAM developments. Ramtron International has worked on this technology for a long time.

PHOTO 5
Cypress Semiconductor's FM6124-QG is a complete SoC. (Photo courtesy of Cypress Semiconductor)

Ramtron was recently acquired by Cypress Semiconductor and currently offers several FRAM devices. Serial and parallel FRAM is available with capacities up to 8 Mb per device.

Photo 4 shows a Cypress 1-Mb parallel access FM28V100. Cypress also offers a low-energy variety of FRAM with active current as low as 3.2 µA.

Cypress even integrated its FRAM module into the FM6124-QG, a complete system-on-a-chip (SoC) that is a standalone event recorder (see **Photo 5**). Although this device is no longer available, its development shows that FRAM blocks can be embedded into digital system ICs to obtain chips that do not require external memory support.

Texas Instruments also offers embedded FRAM products. Its MSP430FR58 and 59 series microcontrollers feature up to 64 KB of integrated FRAM.

A big advantage of using FRAM in microcontrollers is that just one memory can be used for program, data, and information storage instead of having to use separate flash, SRAM, and EEPROM blocks, which has been the trend so far. This makes programming easier and also reduces power consumption. Similar products are also available from Fujitsu.

RESISTIVE RAM

Phase-change memory uses programmed heat-generating current pulses to affect memory cell resistance changes. However, resistive RAM (ReRAM)—a still developing memory breed—uses voltage pulses to make resistance changes. This memory technology utilizes materials and structures where suitable voltages can alter memory cells' resistive states so they can store one or more data bits, similar to PRAM.

There are strong hints that ReRAM is capable of very fast switching with symmetric read and write times of less than 10 ns. This comes with a remarkably low power consumption, which should make this technology ideal for many applications.

As if these attributes were not enough, ReRAM cells are very small and can be placed extremely close together, which results in high-density memory fabrics. Several research

organizations have already demonstrated ReRAM cells smaller than 30 nm × 30 nm. It is easy to see that ReRAM's outstanding features will attract a lot of attention, which is exactly what is happening.

A ReRAM cell is made from a "resistive-switching" material. This is any inorganic or organic material whose electrical resistance can be stably switched between a high and low resistance state by applying a suitable current pulse. Depending on the material involved, the resistance switching can occur because of the formation and disappearance of filamentary conducting paths or due to the movement of individual atoms in the crystal lattice.

Examples of materials that can exhibit this functionality include nickel oxide, tantalum pentoxide, titanium dioxide, and even silicon dioxide. The relevant facts are that the switching action is extremely fast (typically taking place in just a few nanoseconds) and the resistive state is reversible, nonvolatile, and needs very low switching energy.

Figure 4 shows a typical ReRAM cell's structure where the resistive cell stack is located just above the cell selection transistor. M2 and M1 are the top and bottom electrodes

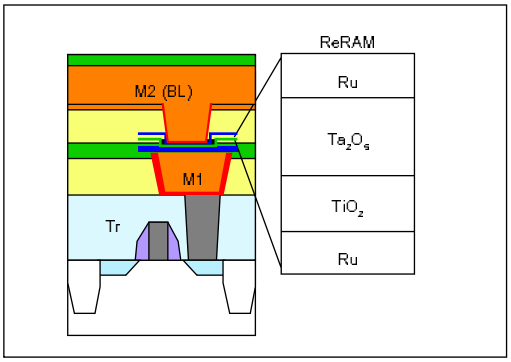


FIGURE 4
A typical resistive RAM cell's structure is shown.

that sandwich the tantalum and titanium oxide-resistive element placed between ruthenium contacts. This design's compact vertical structure reduces the silicon chip area required for each memory cell, making high-density integration possible.

Several companies have announced ReRAM products that are in development or available in low sampling volumes. Micron and Sony have entered into a joint-venture agreement to develop ReRAM chips.

In early 2011, Elpida Memory developed a 64-Mb ReRAM chip using a 50-nm process. The company is working on producing

ABOUT THE AUTHOR

Faiz Rahman (faiz.rahman@electrospell.com) obtained his PhD in Electrical Engineering from Imperial College, London in 1997. After doing postdoctoral work at the University of Nottingham, he worked for the California Institute of Technology on projects based at NASA's Jet Propulsion Lab. Later he served as a senior technology development engineer at Cypress Semiconductor before taking a faculty position at the School of Engineering at the University of Glasgow in 2002. Faiz is currently the Stocker Visiting Professor at Ohio University in Athens, Ohio. His interests include nanotechnology, advanced electronic and optoelectronic devices, organic electronics, and integrated systems.

much higher-density chips later this year. Other companies known to be working on this technology include Samsung, Hynix, Sharp, Fujitsu, and Taiwan Semiconductor Manufacturing.

Panasonic has gone even further by announcing a microprocessor with a built-in ReRAM block. This device boasts power consumption of only a few hundred microwatts and is aimed at low-power, high-battery endurance applications.

Evaluation kits for the Panasonic processor are already available. IMEC, a European semiconductor research organization, has also described the development of high-specification ReRAM ICs at developer conferences. ReRAM's impressive performance metrics are serving to bolster the development of this memory technology. We are certain to see ReRAM ICs becoming widely available two to three years from now.

THE WINNING TECHNOLOGY

A multitude of new memory technologies are vying for influence. Which one or two will emerge as commercially successful products? The answer is not clear despite manufacturers arguing strongly in favor of their respective products. To be commercially viable, any technology needs to meet a set of stringent requirements that include widespread availability and reasonably low cost.

In the case of memory technologies, another important requirement is availability at high-storage densities. This is where most new memory technologies tend to struggle. Memory densities rise incrementally, so it takes many years before a memory type matures enough to be commercially available as a large-capacity chip. What is clear is that some, if not all, of the memory technologies examined in this article will become mainstream in upcoming years. Anticipating their emergence, **Table 1** compares the attributes of these currently known technologies.

Despite some of the hype surrounding these emerging memory technologies, it is reasonable to assert that no one memory will be "universal" (i.e., capable of replacing all other types in a complex application). Rather, each will open up its own niche. However, it is clear that all the new memory technologies have strong performance attributes, which will make them popular and enable higher-performance digital systems. The next few years will see some of these technologies becoming commonplace, further expanding the options available to digital system designers.

circuitcellar.com/ccmaterials

RESOURCES

Buffalo Technology, Inc., www.buffalotech.com

Elpida Memory, Inc., www.elpida.com

Fujitsu, www.fujitsu.com

Hynix Semiconductor Inc., www.hynix.com

IMEC, www.imec.be

Micron Technology, Inc., www.micron.com

Panasonic Corp., www.panasonic.com

Samsung Electronics America, Inc., www.samsung.com

Sharp Electronics Corp., www.sharpusa.com

Sony Corp., www.sony.com

Taiwan Semiconductor Manufacturing Company, Ltd., www.tsmc.com

Toshiba Corp., www.toshiba.com

SOURCES

FM28V100 and FM6124-QG FRAM

Cypress Semiconductor Corp. | www.cypress.com

EMD3D064M MRAM

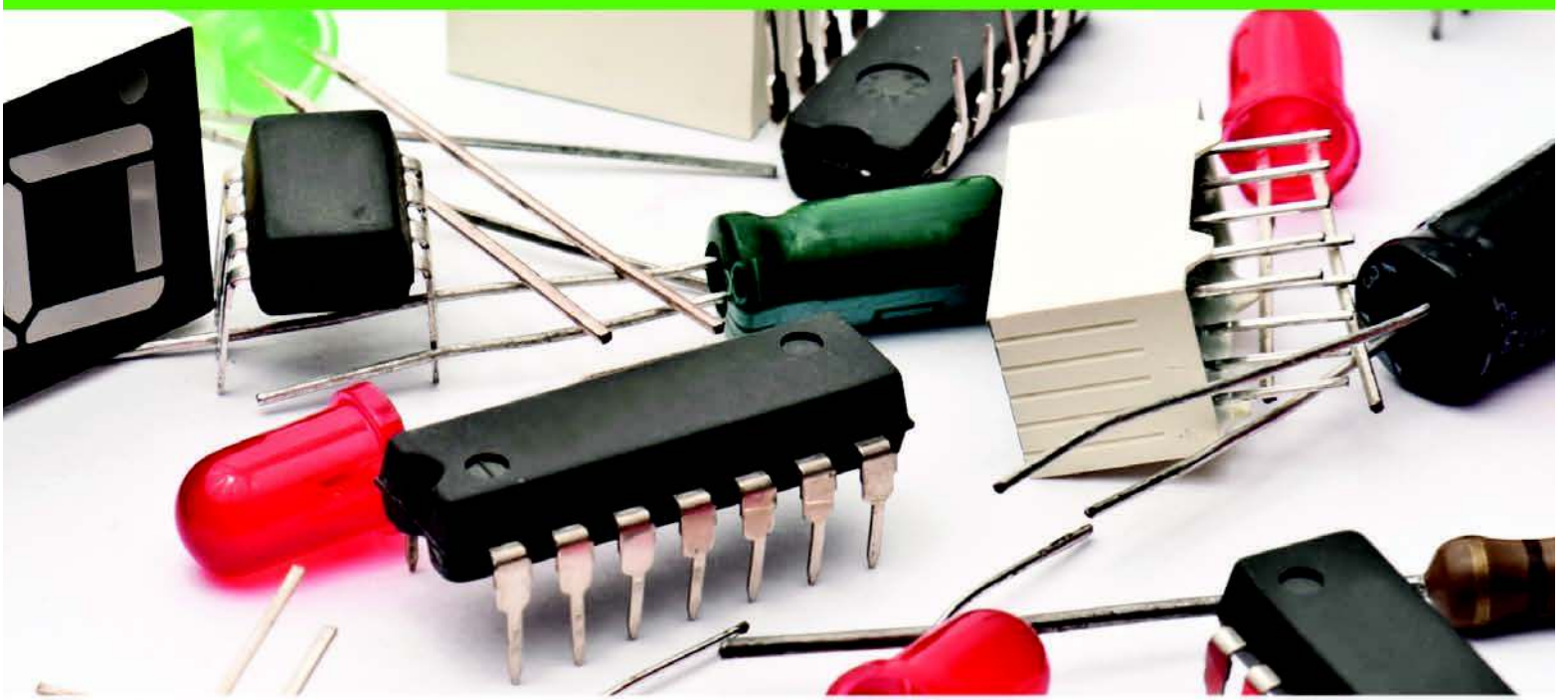
Everspin Technologies, Inc. | www.everspin.com

MSP430FR5969 Wolverine mixed-signal microcontroller

Texas Instruments, Inc. | www.ti.com

A BETTER PLACE FOR ELECTRONICS KITS

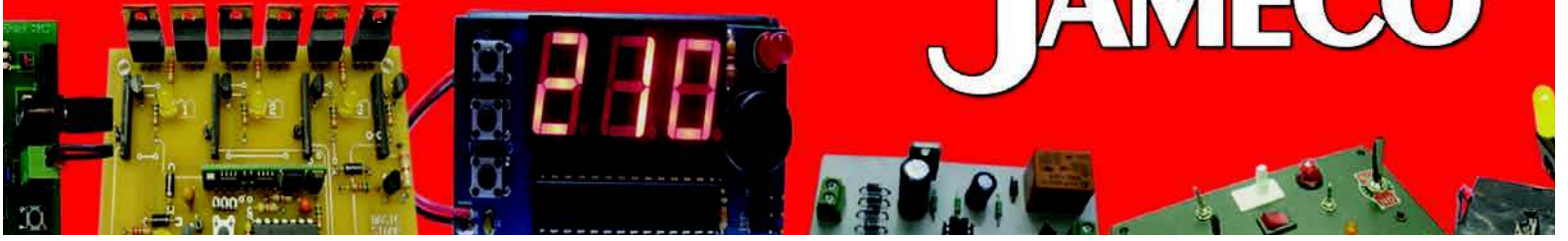
MORE KITS • WILDER KITS • YOUR KITS



- Over 600 electronics projects and kits for sale
- Electronic kits you won't find anywhere else
- All projects include components and instructions
- Design your own kit...We'll build it and pay you on every sale!

Join in the fun and visit us at:
www.ClubJameco.com/Kits

Club JAMECO®



THE CONSUMMATE ENGINEER

Impedance Matching

According to Jacobi's law, impedance matching is an important design consideration, especially in communications and when working with high frequencies. This article shows a practical implementation in the form of an antenna for a garage door opener.

By George Novacek (Canada)

COLUMNS

Around 1840, Moritz von Jacobi postulated a fundamental law of electrical engineering, which has since become known as Jacobi's law. It states that maximum power from a source, when viewed from the output terminals, is obtained when the source's internal impedance and the load impedance match.

Consider the DC circuit shown in **Figure 1a**. Here, both the internal and the load impedances are purely resistive. The generator comprises a DC voltage source V_I and internal resistance R_I. The output terminals are load resistor R_L's terminals.

The blue trace in **Figure 1b** shows the power transferred in relation to the R_I:R_L ratio. Maximum power is being transferred only when the two resistors are equal. Because equal current flows through the two equal resistors R_I and R_L, 50% of the power is dissipated by each resistor. Since one half of the power is wasted in the source's internal resistance, the power transfer efficiency is a mere 50%. This has sometimes led to an incorrect conclusion that no more than 50% efficiency can be achieved in electrical circuits.

This is not what Jacobi's law means. It tells us how to choose a load resistor R_L for maximum power transfer, given the internal resistance R_I, but is not intended for selection of the internal resistance R_I, given the load R_L.

Efficiency η (Greek letter eta), of this simple circuit is shown by the red trace in **Figure 1b**, which plots the efficiency η with respect to the load and internal resistance ratio. It is calculated as:

$$\eta = \frac{R_L}{R_L + R_I} = \frac{1}{1 + \frac{R_I}{R_L}}$$

For maximum efficiency, R_I should be as small as possible when compared to R_L to minimize loss in the generator. To visualize the concept, consider three fundamental conditions: First, $R_L = R_I$. As we have already seen, $\eta = 0.5$ (i.e., 50%). If $R_L = \infty$ or $R_I = 0$, $\eta = 1$ (i.e., 100%). And if $R_L = 0$, $\eta = 0$. In other words, with decreasing R_I and/or increasing R_L, η is approaching 100%.

When $R_L = 0$, which is a short circuit, the efficiency becomes zero, as all the power is wasted in R_I. While the efficiency grows with

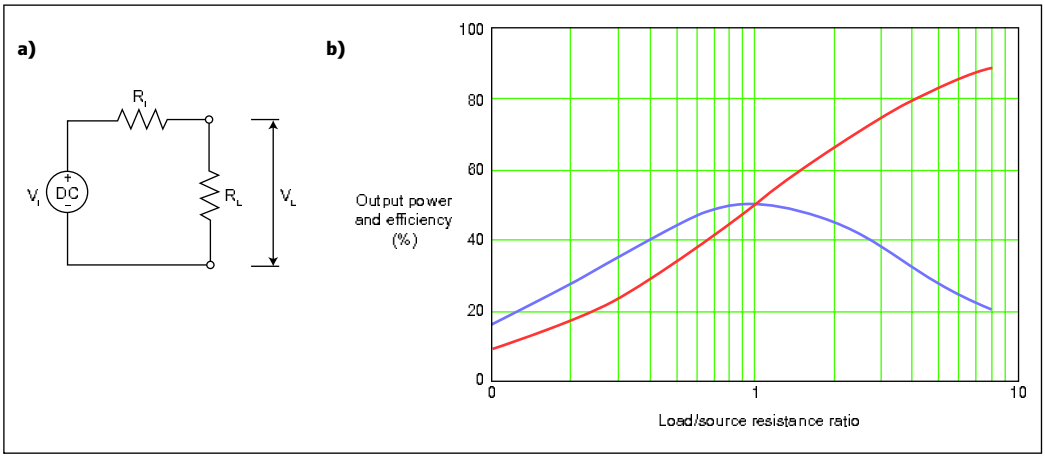


FIGURE 1
a—The generator with its internal resistance R_I and the load resistance R_L is shown. **b**—The blue trace is the plot of the power dissipated by R_L vs. R_I when R_I is constant. The red trace shows the transfer efficiency dependence on an R_I:R_L ratio.

increasing R_L and/or decreasing R_I, the total amount of power transferred decreases on both sides of the R_I = RL condition.

DC POWER TRANSFER

There aren't many situations where a DC power transfer is an issue. However, it is an important design aspect when working with AC signals, especially at higher frequencies. The concept remains the same, except the resistances are replaced with complex impedances.

Once again, for maximum power transfer $Z_I = Z_L$, which can be written as:

$$R_I \pm jX_I = R_L \pm jX_L$$

The resistive components must match, $R_I = R_L$, while the reactive components must cancel each other, that is $X_I = -X_L$. When this happens, the circuit resonates at the following frequency:

$$f = \frac{1}{2\pi\sqrt{LC}}$$

MATCHING REQUIREMENTS

This is when, things become interesting—particularly in communications, be it digital (e.g., the USB protocol), analog, or RF. Even a PCB layout needs to respect impedance-matching requirements.

Take a microprocessor running at a relatively slow, 100-MHz clock speed. One requirement for a reliable digital operation is a crisp clock (i.e., a rectangular clock waveform with short rise and fall times).

A rectangular wave is composed of its fundamental frequency and several of its odd harmonics. Generally, all odd harmonics up to about the 15th are needed for a reasonably crisp clock.

Now, let's do the math: The 100-MHz clock's 15th harmonic is 1,500 MHz. Its wavelength λ (Greek letter lambda) is approximately 200 mm (199.862 mm to be precise).

A PCB trace or a conductor with a length exceeding λ/8 begins to behave as a transmission line. In our example, this would be 25 mm, or a little less than 1". Such a length is easily exceeded if the clock is distributed to several ICs. When a connection becomes a transmission line, it exhibits a characteristic impedance based on its physical properties and the line must be terminated at both ends.

The source and the load impedances must match that of the transmission line. This is not just to ensure maximum power transfer—which may not be needed—but if the impedances are not matched, standing waves develop along the line as some of the power is reflected back. This can cause unwanted electromagnetic interference (EMI) and the signal itself can become corrupted. Physical damage to the hardware can even result from a serious impedance mismatch. The match's degree is expressed as a standing wave ratio (SWR):

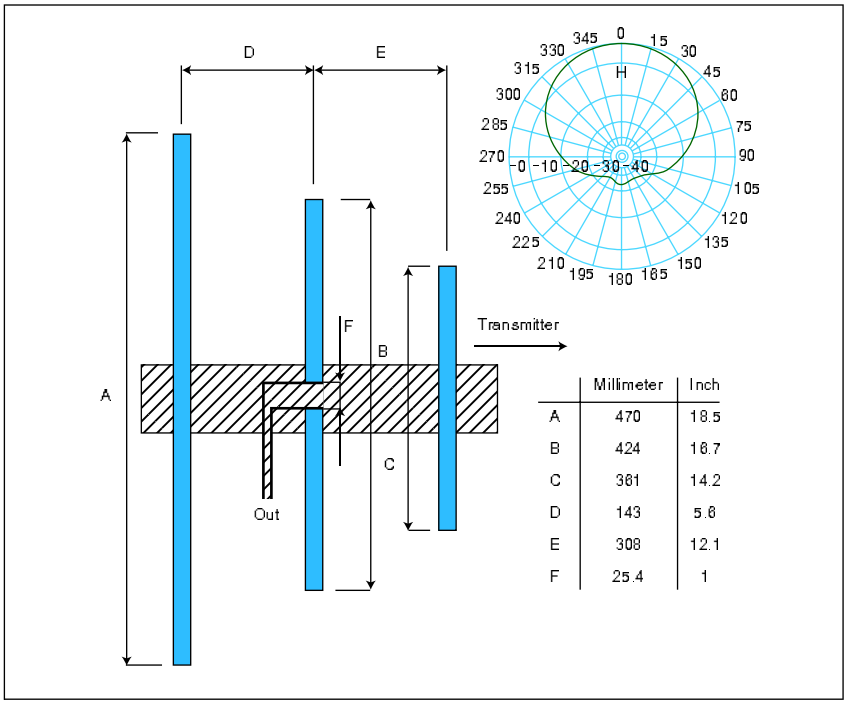
$$SWR = \frac{Z_L}{Z_0} \text{ for } Z_L \geq Z_0$$

and

$$SWR = \frac{Z_0}{Z_L} \text{ for } Z_0 \geq Z_L$$

SWR = 1 is perfect, but SWR <= 2 is considered a good match. Instead of SWR, standing wave voltage ratio (SWVR) is often used because it can be easily measured.

Matched impedance equipment is used every day (e.g., telephones, computer interface cables, Ethernet, serial communications cables, and RF coaxial or twin lead cables for TV distribution). Each interface cable class is designed to have a certain impedance to make



	Millimeter	Inch
A	470	18.5
B	424	16.7
C	361	14.2
D	143	5.6
E	308	12.1
F	25.4	1

FIGURE 2
This is the dipole antenna for 315 MHz (not to scale).

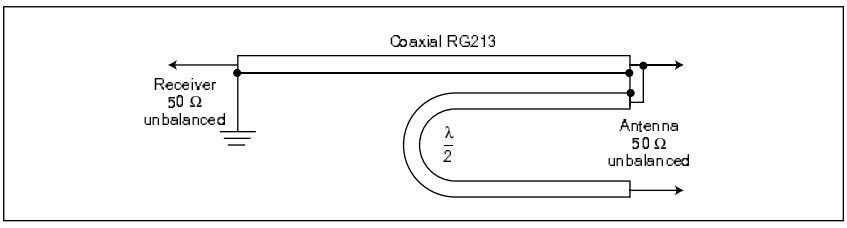


FIGURE 3
A 1:1 50-Ω balun made out of a coaxial cable is shown.

COLUMNS

ABOUT THE AUTHOR

George Novacek (gnovacek@nexicom.net) is a professional engineer with a degree in Cybernetics and Closed-Loop Control. Now retired, he was most recently president of a multinational manufacturer for embedded control systems for aerospace applications. George wrote 26 feature articles for *Circuit Cellar* between 1999 and 2004.

its matching universal for given applications. For instance, with RF work, a 50-Ω impedance is conventional.

Impedance matching methods are beyond the scope of this article, but I will mention that resistors and transformers are used to match resistance while networks with capacitors and/or inductors are used to cancel the reactive components.

A PRACTICAL EXAMPLE

Let's consider a practical impedance matching example. My garage door opener's operating range along my approximately 150′ (46-m) long driveway was poor and inconsistent. Both the transmitter and the receiver use surface acoustic wave (SAW) resonators to control their frequency, so the poor range was not due to detuning.

Increasing the transmitter power to extend the range was not an option. I figured a combination of the signal absorption plus multipath distortion were the most likely culprits of the poor range. I decided to build a good antenna for the receiver to fix the problem.

The original antenna was a piece of a λ/4 long wire that was around 9.4″ (238 mm) for its 315-MHz operating frequency. This indicated the receiver's input impedance to be about 50 Ω (the theoretical impedance of a monopole λ/4 antenna is 36 Ω). I decided to replace it with a three-element Yagi antenna to increase the gain and directionality. I used a free YagiCAD program I had used in other projects for the design.

Figure 2 shows the antenna. The driven element is a dipole with about 50-Ω impedance (theoretical dipole impedance is 73 Ω). The calculated gain is 7.7 dB with directionality in the horizontal plane also shown in **Figure 2**. The antenna gain is expressed as a ratio of its output to a theoretical isotropic antenna with a spherical radiation pattern. The gain is achieved by focusing the beam, just like putting a reflector behind a light bulb.

I built the antenna with 0.5″ plumbing copper pipe because it is easy to work with. At 315 MHz, the skin effect causes the surface current penetration of a mere 3.67 µm, so just about any stiff conductor would work. To keep the construction simple, I used a piece of wood for the boom and placed the antenna in the rafters above the garage.

RG213 coaxial cable is a good choice for connection to the receiver. It is popular among Ham amateurs due to its characteristics (i.e., 50-Ω impedance, reasonable price, and low loss).

The receiver and the dipole are good match for the 50-Ω coaxial cable. The only problem is that the dipole's impedance is balanced while the coaxial cable and the receiver impedances are unbalanced. The solution is to use a 1:1 balun (i.e., a balanced-to-unbalanced transformer). You can purchase it, but it's easy to make one (see **Figure 3**). Just add a loop of the coaxial λ/2 long, which causes 180° phase shift of the signal and, *voilá*, you have symmetry. Half the wave length of a 315-MHz signal is 18.74″ (476 mm).

With a coaxial cable, the RF wave does not propagate as fast as in vacuum; therefore you must take the coaxial cable type's specific propagation velocity factor into account. For RG213, it is 66%. Consequently, the loop's length will be 12.4″ (314 mm). With the antenna installed, the garage door opener range increased significantly. The door can now be consistently operated from the road.

A GOOD UNDERSTANDING

We encounter impedance matching issues daily, especially when interfacing a diversity of electronic equipment. Most of the time, we purchase hardware with proper cables and terminations without even realizing impedance matching was designed in. Nevertheless, it's always good to understand what goes on behind the scenes.

circuitcellar.com/ccmaterials

RESOURCES

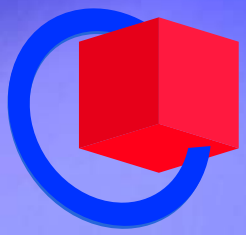
J. Wetherall, University of San Diego, "Impedance Matching Network Designer," 2001.

Wikipedia, "Square Wave Frequency Spectrum Animation."

YagiCAD, Antenna Design Software, www.yagicad.com.

Register now and secure your ticket!
embedded-world.de

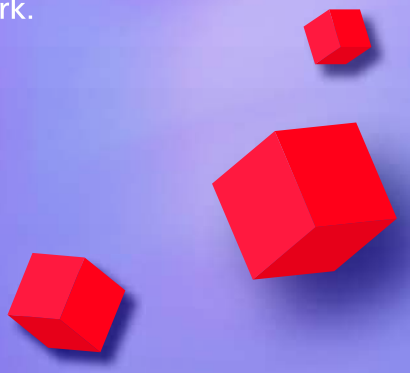
Nürnberg, Germany
February 25 – 27, 2014



embedded world 2014
Exhibition&Conference
... it´s a smarter world

Summit for Innovations

The world's largest event for Embedded-Technologies is inviting you. The international Embedded-Community trends of tomorrow will be set right here! A must attend event, to establish contacts and expand your network.



Media Partners

elektroniknet.de
computer-automation.de
energie-und-technik.de
karriere-ing.de
Markt&Technik
Elektronik
Elektronik automotive
ENERGIE & TECHNIK
Computer & AUTOMATION
MEDIZIN elektronik
DESIGN & ELEKTRONIK KNOW-HOW FÜR ENTWICKLER

Organizers

Trade Show
NürnbergMesse GmbH
Tel +49 (0) 9 11.86 06-49 12
visitorservice@nuernbergmesse.de

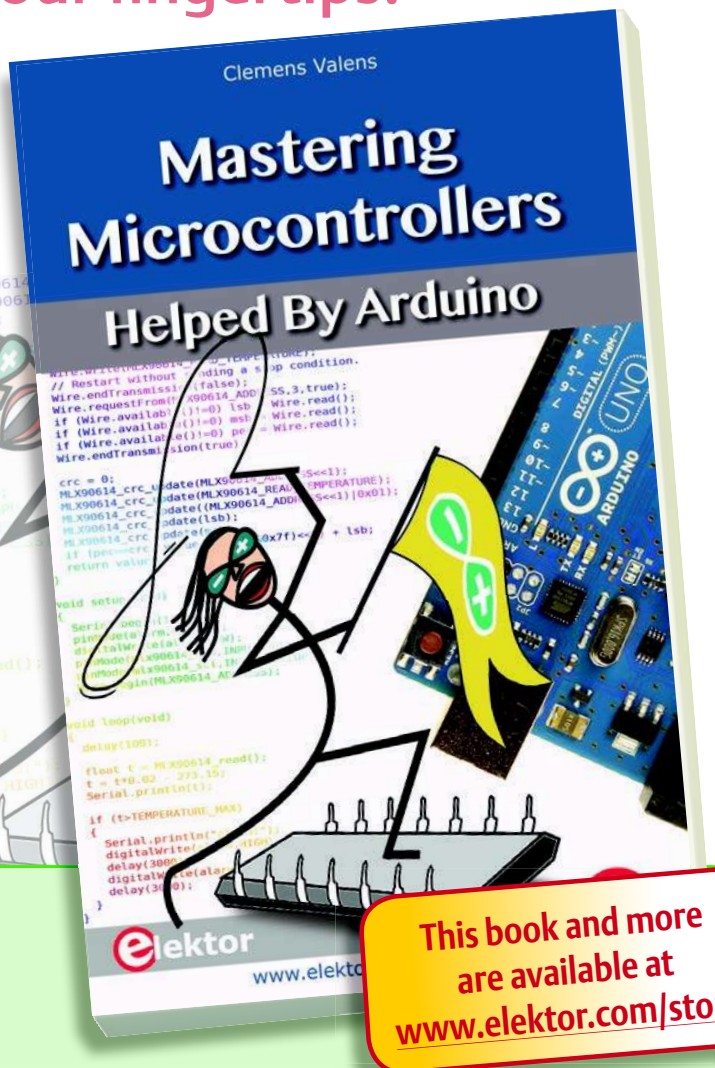
Conference
WEKA FACHMEDIEN GmbH
Tel +49 (0) 89.2 55 56-13 49
info@embedded-world.eu

NÜRNBERG MESSE

BOARDS, BOOKS, DVDs AND MORE AT WWW.ELEKTOR.COM/STORE

Elektor.STORE

The world of electronics at your fingertips!



This book and more are available at www.elektor.com/store

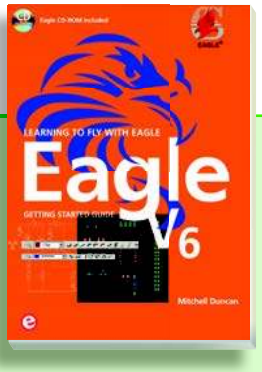
Books

Very Original Applications!

Mastering Microcontrollers Helped by Arduino

Arduino boards have taken the world by storm since their initial development. The aim of this book is not only to let you enter the World of Arduino, but also to help you emerge victorious and continue your microcontroller programming learning experience by yourself. In this book theory is put into practice on an Arduino (uno) board using the Arduino programming environment. Having completed this fun and playful course, you will be able to program any microcontroller, tackling and mastering I/O, memory, interrupts, communication (serial, I²C, SPI, 1-wire, SMBus), A/D converter, and more. This book will be your first book about microcontrollers with a happy ending.

348 pages • ISBN 978-1-907920-23-3 • $56.40



Learning to fly with Eagle

Eagle V6 Getting Started Guide

The book is intended for anyone who wants an introduction to the capabilities of the CadSoft's Eagle PCB design software package. The reader may be a novice at PCB design or a professional wanting to learn about Eagle , with the intention of migrating from another CAD package. After reading this book while practicing some of the examples, and completing the projects, you should feel confident about taking on more challenging endeavors. This book is supplied with a free copy of Eagle on CD-ROM for MS Windows, Linux and Mac.

208 pages • ISBN 978-1-907920-20-2 • $47.60

Programming step-by-step

Android Apps

This book is an introduction to programming apps for Android devices. The operation of the Android system is explained in a step by step way, aiming to show how personal applications can be programmed. A wide variety of applications is presented based on a solid number of hands-on examples, covering anything from simple math programs, reading sensors and GPS data, right up to programming for advanced Internet applications. Besides writing applications in the Java programming language, this book also explains how apps can be programmed using Javascript or PHP scripts.

244 pages • ISBN 978-1-907920-15-8 • $56.40

Prices and item descriptions subject to change. E. & O.E

Books

Theory and Practice

Practical Digital Signal Processing using Microcontrollers

This text on DSP reflects the growing importance of discrete time signals and their use in everyday microcontroller based systems. The author presents the basic theory of DSP with minimum mathematical treatment and teaches the reader how to design and implement DSP algorithms using popular PIC microcontrollers. The author's approach is practical and the book is backed with many worked examples and tested and working microcontroller programs

428 pages • ISBN 978-1-907920-21-9 • $72.50

MIFARE and Contactless Cards in Application

RFID

MIFARE is the most widely used RFID technology, and this book provides a practical and comprehensive introduction to it. Among other things, the initial chapters cover physical fundamentals, relevant standards, RFID antenna design, security considerations and cryptography. The complete design of a reader's hardware and software is described in detail. The reader's firmware and the associated PC software support programming using any .NET language.

484 pages • ISBN 978-1-907920-14-1 • $72.50

Embedded Linux Made Easy

Today Linux can be found running on all sorts of devices, even coffee machines. Many electronics enthusiasts will be keen to use Linux as the basis of a new microcontroller project, but the apparent complexity of the operating system and the high price of development boards has been a hurdle. Here Elektor solves both these problems, with a beginners' course accompanied by a compact and inexpensive populated and tested circuit board. This board includes everything necessary for a modern embedded project: a USB interface, an SD card connection and various other expansion options. It is also easy to hook the board up to an Ethernet network.

Populated and tested Elektor Linux Board

Art.# 120026-91 • $93.30

Elektor is more than just your favorite electronics magazine. It's your one-stop shop for Elektor Books, CDs, DVDs, Kits & Modules and much more!

www.elektor.com/store

Elektor US
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
Fax: 860.461.0450
E-mail: order@elektor.com

Android Elektorcardioscope

Instructive, fascinating, and potentially useful to everyone: perform your own electrocardiograms on your Android smartphone or tablet! The project involves skillfully combining a small PIC interface to control an analog input stage with a great deal of software. Our ECG interface is available in the form of a ready-to-use module to which you just have to add four electrodes and an Android application for smartphone or tablet; there's no physical connection between this terminal and the interface, as it uses Bluetooth communication!

Ready assembled board

Art.# 120107-91 • $132.00

500 ppm LCR Meter

The remarkable precision of this device and its amazing ease of use are the result of careful design. It works so well behind its uncluttered front panel that one could almost forget the subtleties of the measurement techniques employed. A dream opportunity for our readers who are passionate about measurement to enjoy themselves. If, like us, you wonder at the marvels modern techniques bring within our reach, come along and feel the tiny fraction of a volt.

Set: Main board and LCD board, assembled and tested

Art.# 110758-93 • $300.00

Kits & Modules

EMBEDDED IN THIN SLICES

Embedded File Systems (Part 2)

File System Integrity

The first part of this article series introduced the topic of Linux embedded file systems. This article discusses file system integrity across power outages and system crashes and describes some integrity assurance solutions.

By Bob Japenga (USA)

COLUMNS

In 1975, I was working on a PDP-11 system that used the RT-11 file system. I noticed that the disk access became slower over time. You could issue the `SQUeeze` command to the file system (all 2.5 MB of it), which would resolve this problem. Of course, the command took almost a day to complete.

PCs had a similar problem caused by disk fragmentation that prompted many people to buy a new PC when their old one moved at a sluggish pace. They thought it was worn out. Instead the disk was just fragmented. Disk fragmentation happens any time one file's data is located in noncontiguous physical locations on the disk.

"It is important for you as the designer to know exactly what your file system can and cannot do. Can it be powered down 10,000 times without corrupting data?"

Once when my dentist complained about how slow his PC had become, he asked me for recommendations for a new PC. I suggested he use a defrag program instead of replacing his PC. He was very grateful—but he still drills me now and again.

STRUCTURE MATTERS

Both problems were caused by the file system's architecture. File systems have a physical disk structure that defines where the data is stored in the physical media (e.g., hard drive, floppy, USB stick, or flash drive). The logical disk structure defines how files are stored on the disk. The way in which the design maps the logical disk structure to the physical disk structure dramatically affects any file system's performance and reliability.

For example, neither the Macintosh file system (after OS 10.2) nor Linux systems need defragging. Both have file structures that do not require this operation.

I bring this up because most people are familiar with older Windows systems' defragging requirements. Most don't know that this is not inherent in all file systems but is due to a design decision about the file system's structure. The file system's structure can affect things such as robustness over time, excessive disk space usage, and so forth.

This brings us to another feature critical to embedded systems: robustness across unexpected power outages or system crashes. If you are using a file system in an embedded system, you don't want it to ever get corrupted across a power outage or crash! Some file systems prevent this from happening by requiring an orderly shutdown. With embedded systems, you don't want to require an orderly shutdown to be performed before you power down.

One embedded Linux single-board computer (SBC) provider continues to provide misinformation about this on its website. It insists that embedded systems must provide an orderly Linux shutdown to prevent data corruption. Really? This depends on what Linux file system you are using. Thus, it is important for you as the designer to know exactly what your file system can and cannot do. Can it be powered down 10,000 times without corrupting the data?

One of the top RTOS developers didn't do this on a system we designed and it required us to retrofit thousands of devices already in the field with a battery backup to prevent

unrecoverable data corruption following inadvertent loss of power.

In the same way smart file system architecture can eliminate the need for defragging, your file system's architecture can affect whether or not a file system's integrity can be guaranteed across every power-down scenario. Let's start by looking at the problem.

THE POWER-DOWN PROBLEM

Early on, when using SD cards and compact flash in cameras and other devices, many users discovered that the inexpensive cards could get corrupted when removing the card while writing to it. When the cards became corrupted, you could lose all of your data on the card.

Even today, my new Nikon camera has strict warnings: *Do not remove the card while the access light is on*. If you ever had an SD card, compact flash, or memory stick become corrupted and lose all of its data (your pictures, songs, etc.), you know how frustrating that can be.

Imagine if my company put this sticker on a embedded medical device we designed to go inside the human body: *Don't remove power without first performing an orderly shutdown.* They would probably call for a medical orderly after they shut us down. Embedded systems usually cannot control when power is going to be removed. And not every system can be held up long enough to provide time to perform an orderly shutdown.

If all the information about a file could be atomically written to one location in the file system, powering down while writing to that one location would be acceptable. Only the most recent data would be lost. Other parts of the file system would not be corrupted.

But real file systems don't keep all of the information about a file in one place. That would make accessing the file very time consuming. Imagine having to look across every block of a terabyte drive trying to find your unique file. As a minimum, file systems usually create other information about the file that they store in another place.

One example of this is directory information, which tells the system where the file is located. This makes finding your file faster. File systems also want to store other information, such as who has permission to read, write, delete, and append the file; who owns the file; when the file was last modified; when the file was created; when the file was last accessed; and pointers to the media's physical blocks (hard drive or flash) where the file is kept. This is called metadata. Information about most files is contained in at least two locations: the directory and the file itself.

Therein lies the problem. If you write the data successfully and then start updating the directory when the power goes down, you run the risk of corrupting the directory entries for other files. This is unacceptable for embedded systems. You want a file system that can go through millions of shutdowns during file writes and never corrupt either the existing file being written or any other file in the system.

THE SOLUTION

Journaling file systems came to the rescue. The idea for such architecture was conceived in the 1980s. IN 1990, IBM released the first Journaled File System, which it called JFS.

The basic concept is that the file system creates a journal entry of all changes that are going to be made when a file is written or is changed. Thus, in case of an incomplete operation, it can quickly and easily back out the last change. If the power goes down or something causes the software to crash, the file system will not be corrupted and it can quickly recover. The file system can read the journal to recover the system to the point it was at just before the write was commanded.

"If you write the data successfully and then start updating the directory when the power goes down, you run the risk of corrupting the directory entries for other files. This is unacceptable for embedded systems."

Some call journaling file systems log-structured file systems since the difference between keeping a log and keeping a journal is pretty minute. It is beyond the scope of this thin slice, but for argument's sake, let's consider log-structured file systems of the same ilk as journaling file systems.

LINUX FLASH-BASED JOURNALING FILE SYSTEMS

An embedded systems designer using Linux has several choices to make when selecting a file system. And the implications can be significant. Explanations of some of the journaling file systems we have used follows:

Journaling Flash File System V. 2 (JFFS2)—This has been the industry benchmark for flash-based journaling file systems since it was the first one introduced into the mainstream Linux distribution. JFFS2 has compression built into its core. It also has wear leveling built into the structure.

We have worked mostly with JFFS2. It has several quirks we have discovered over

ABOUT THE AUTHOR

Bob Japenga has been designing embedded systems since 1973. In 1988, along with his best friend, he started MicroTools, which specializes in creating a variety of real-time embedded systems. With a combined embedded systems experience base of more than 200 years, they love to tackle impossible problems together. Bob has been awarded 11 patents in many areas of embedded systems and motion control. You can reach him at rjapenga@microtoolsinc.com.

time. One is that, because of compression, you cannot tell exactly how much disk space you have left as you run low. For example, if there is 1 MB of raw disk space left and you have 2 MB of data to write, can you commit the write? You don't know unless you know the esoteric compression algorithm efficiency on the particular data you want to write. We have worked around this algorithmically but it is nontrivial.

In addition, it can have a fairly long boot time on large (i.e., greater than 32 MB) flash drives. It also uses significantly more RAM than the other options. For us, JFFS2's primary disadvantage is with larger flash drives. We are not convinced it works well on systems above 64 MB.

Yet Another Flash File System V. 2 (YAFFS2)—In 1975, a friend of mine kept going on and on about YACC (yet another compiler compiler). I didn't even know why you needed a compiler to compile compiler code! Around 2001, the YAFFS2 creators reached back into that bit of history to create YAFFS.

We are currently using YAFFS2 on one system. Under our tests, it has proven to be very robust across thousands of power failures occurring during disk writes. According to the Yaffs website, it has been crash tested hundreds of thousands of times. Due to its architecture, YAFFS2 boots up much more quickly on large drives than JFFS2. YAFFS2 takes significantly longer to remove a file than the other options.

We have very limited experience with quantities of these in the field so talk to me in a few years. We "chose" it because it came with the SBC we were using. Unlike JFFS2, YAFFS does not contain built-in compression.

Unsorted Block Image File System (UBIFS)—Considered the successor to JFFS2 because it contains journaling and compression, we seamlessly switched to UBIFS once on a project when JFFS2 was taking too long to boot. It reduced our boot time by 30 s! In most benchmark tests, it is faster than the other file systems in reads and writes and average on file deletions. We have had no problems using UBIFS. We have hundreds of thousands of JFFS2 systems in the field and comparatively few UBIFS. Again, talk to me in a few years

LogFS—In 2010, yet another flash file system (LogFS) was introduced into the mainstream Linux distribution. One of the problems with JFFS2 is that it keeps the inode tree (think of this as metadata handling where the logical blocks map to the physical blocks) in RAM so it must create this every time you boot. LogFS keeps this on disk. Clearly this creates some performance degradation, but we do not have any experience with this file system.

CHOOSING A JOURNALING FILE SYSTEM

Some choices are made for you with the hardware. Not all file systems work with all the different kinds of flash. Sometimes you design around a SBC with a flash file system already installed. Yes you could change it—but do you need to?

JFFS2 and UBIFS have built-in compression, which may drive your decision if you are concerned about how little disk space you have. Some are of the school that newer is better. Some like to stick with what they know and has proven robust. We have had just enough problems with JFFS2 that I tend to favor working with a later design (e.g., UBIFS). Benchmarks on UBIFS look very good (see Resources for more information).

CHOICES, CHOICES

As embedded systems designers, we sometimes wish we could have fewer options when choosing a processor, a RTOS, a file system, a development toolchain, and so forth. But the fact remains that we do have many choices.

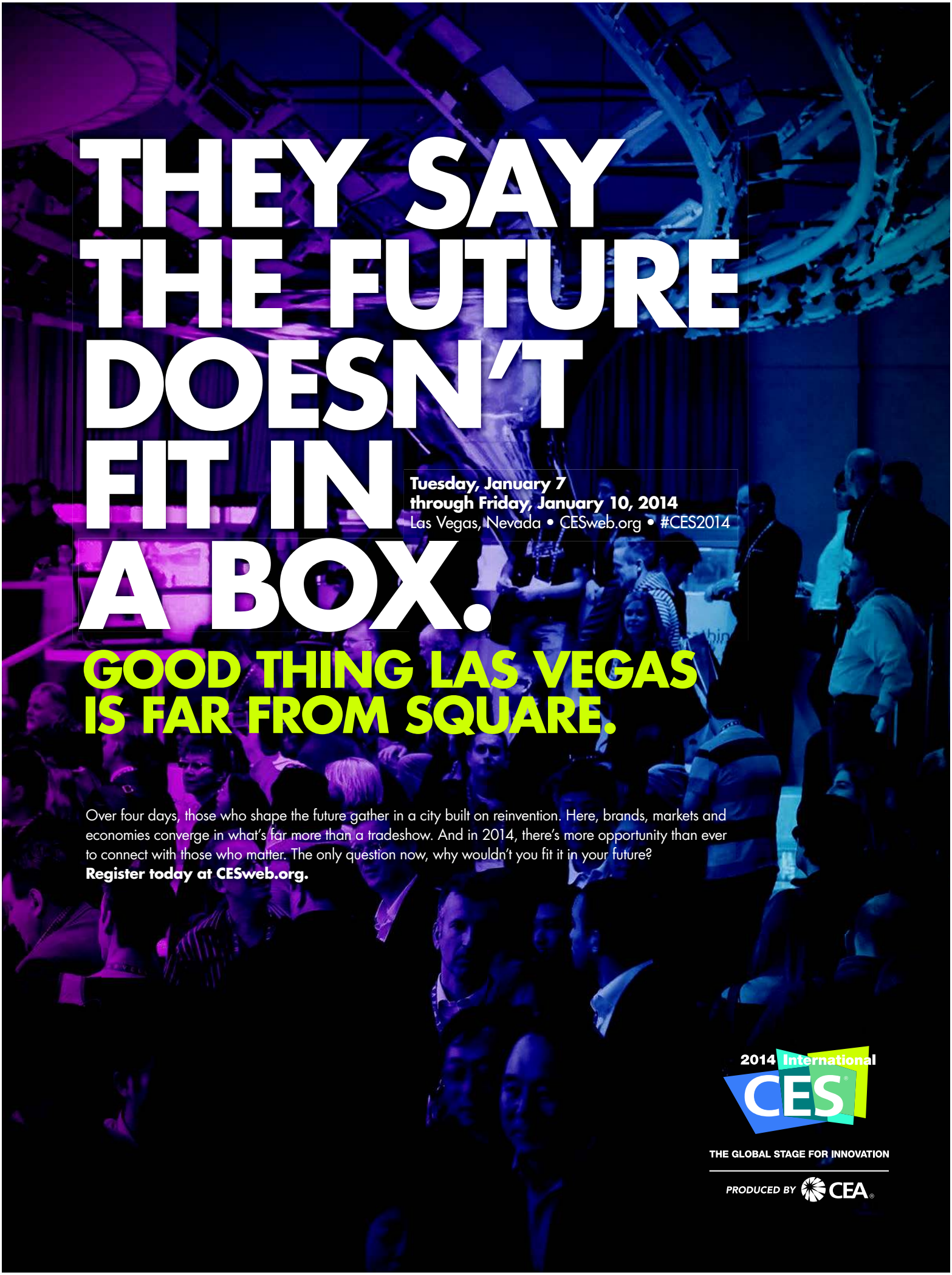
Understanding the implications of the choices will help us create more robust systems. Choosing a flash file system for your embedded Linux system is important. Hopefully I have introduced some options so you can take this beyond thin slices.

circuitcellar.com/ccmaterials

RESOURCES

eLinux.org, "Flash Filesystem Benchmarks 3.1."

Yaffs, open-source file system, www.yaffs.net.


THEY SAY THE FUTURE DOESN'T FIT IN A BOX.
Tuesday, January 7
through Friday, January 10, 2014
Las Vegas, Nevada • CESweb.org • #CES2014
GOOD THING LAS VEGAS IS FAR FROM SQUARE.
Over four days, those who shape the future gather in a city built on reinvention. Here, brands, markets and economies converge in what's far more than a tradeshow. And in 2014, there's more opportunity than ever to connect with those who matter. The only question now, why wouldn't you fit it in your future?
Register today at CESweb.org.
2014 International CES
THE GLOBAL STAGE FOR INNOVATION
PRODUCED BY CEA

PROGRAMMABLE LOGIC IN PRACTICE

Connecting FPGA Hardware to Virtual Test Benches

FPGA simulators are important when validating and debugging designs. But sometimes a critical design piece only exists in hardware, which limits testing to the physical device. This article introduces hardware co-simulation, which provides the flexibility and power of a computer-based simulation, but connects to a real module running on your physical FPGA.

By Colin O'Flynn (Canada)

COLUMNS

In my last article, I introduced the concept of using logic analyzers built into your hardware design to debug and validate hardware implementations. In this article, I'd like to show you to something even more powerful: controlling hardware modules in an almost arbitrary fashion from your computer. This can greatly simplify your debugging and validation process.

The time you invest to learn these techniques will quickly come back to you in increased productivity. As always, detailed project files, example videos, and additional instructions are posted on the companion website, ProgrammableLogicInPractice.com.

HARDWARE CO-SIMULATION

Design simulation is almost always required for functional testing. Simulations enable rapid code iterations. You can easily probe into any network without lengthy synthesis/implementation phases.

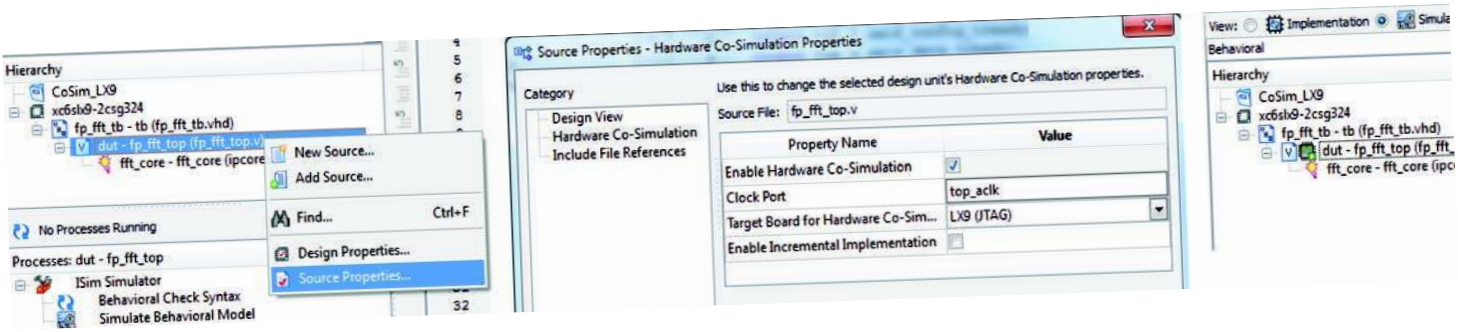
There are two common downsides to simulation. First, in complex designs, it can take a long time to run the simulation. Second, your design may interface to external hardware, which you cannot accurately simulate.

Hardware co-simulation (HCS) is often the answer to both of these problems. In this article, I'll describe using the Xilinx ISE toolchain, for which HCS is supported by the WebPack (free) version of the tools. With HCS, a module is pushed into your real FPGA, which communicates with the rest of the software-based simulation over the JTAG connection.

There are some basic limitations: the clock to the module is not a predictable frequency, you can only select a single module to implement in hardware and you cannot push the topmost module to hardware. However, the single module chosen can have submodules. If you want to push your design's topmost module into hardware, you will typically just need a wrapper.

If you are using HCS to increase the simulation speed of certain operations, the lack of a constant predictable clock isn't normally an issue. For example, with a fast Fourier transform (FFT) operation, you may wish to push this into hardware. The time to

PHOTO 1
Enabling hardware co-simulation is a simple matter of selecting the module, authorizing it, and defining the interface and clock. You may also need to specify a constraint (UCF) file if you're interfacing to external hardware.



download parameters, run the operation, and upload results back to the computer may be much faster than running the simulation on the computer.

I think HCS's more interesting use is where your hardware interfaces to the real world. Perhaps you are using the memory controller block or a Xilinx PCI Express block in the FPGA and want to test your code with the exact external hardware. The example I'll use builds on a simple serial block where the serial blocks are connected to a real hardware serial port.

A specific example of this may be when you are designing a serial protocol. You are developing the software API, which passes messages over the serial port to your hardware board. With HCS, you can use the real API developed with the physical hardware, send the messages out the serial port, receive them on the physical board, and then pass the messages into the FPGA simulator (see **Figure 1**).

Any output from the FPGA simulation is sent out the real physical port where the other device is listening. Is it crazy to take the messages from the computer and pass them out to hardware, only to bring them back into the computer? I don't think so, because this flow enables you to test the system in a realistic situation, making your verification and debugging job easier. Since the code is running in simulation, you aren't limited to only probing specific signals, as with an integrated logic analyzer (ILA). Of course, your FPGA may connect to other devices (e.g., a microcontroller) where, without HCS, it would be almost impossible to debug!

As an example, I ran Xilinx's FFT tutorial (Application Note UG817, see Resources) on the Avnet LX9 MicroBoard. You'll have to recreate the FFT core and adjust the number of samples to fit the LX9 (see detailed instructions at ProgrammableLogicInPractice.com). At this point, you could just run the FFT as a regular software simulation, but you'll want to push the FFT operation down to hardware instead.

The HCS needs you to define a board-support package (BSP) for your hardware. Basically you need to specify a clock source's location. The HCS module will use this pin for its own clocking purpose. This clock will be different from any other clock your module may be using. The tools usually won't work well if you reuse a clock.

From your test bench, you will normally input a clock into the unit under test (UUT). HCS treats this clock in a special manner and will generate it from the BSP-specified clock. This clock will be under software control and won't run in a regular manner. In this FFT example, I measured the test bench clock as having a 50-ns pulse width, but pulses occur at a 40-to-1,100-Hz rate, depending on the interface.

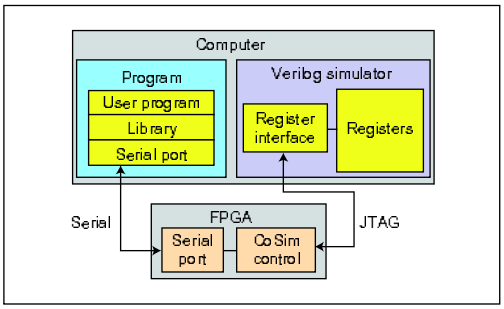


FIGURE 1
Developing a computer interface for your FPGA? Why not use hardware co-simulation to enable your computer program to talk to your FPGA simulator. You are using the real hardware interface, but giving yourself the debugging and verification power a software simulator provides.

To set up the BSP, find the hwcosim.bsp in your ISE installation. In my ISE installation, it is located at C:\Xilinx\14.4\ISE_DS\ISE\sysgen\hwcosim\data\hwcosim.bsp. **Listing 1** shows an example of the BSP format to add support for the Avnet LX9 MicroBoard.

With this file modified, you can launch (or relaunch) ISE. By selecting the appropriate board, you can enable the "Hardware Co-Simulation" option for the device under test (DUT). **Photo 1** shows the process and resulting icon indicating that hwcosim will be used. At this point, you simply select the test bench module (fp_fft_tb in **Photo 1**) and select "Simulate Behavioral Model" as normal.

SPEED-UP RESULTS

Note that the simulation's speed will be highly dependent on the JTAG cable speed. The FFT's software-only simulation takes 47.8 s. Using the Avnet LX9 MicroBoard with a built-in USB-JTAG cable, the simulation takes 150 s and the test bench clock runs at between 40 and 70 Hz. Using a Xilinx Platform Cable USB on the same board means the simulation takes only 11.2 s and the test bench clock runs at between 800 and 1,100 Hz. In this example, you need the Platform Cable USB's additional

```
'lx9-jtag' => {
  'Description' => 'LX9 (JTAG)',
  'Vendor' => 'Xilinx',
  'Type' => 'jtag',
  'Part' => 'xc6slx9-2csg324',
  'Clock' => [
    {
      'Period' => 10,
      'VariablePeriods' => [ 15, 20, 30 ],
      'Pin' => 'C10',
    },
  ],
  'BoundaryScanPosition' => 1,
},
```

Listing 1
You'll have to add this code to the board support package for the hardware co-simulation. On my computer, this was installed at C:\Xilinx\14.4\ISE_DS\ISE\sysgen\hwcosim\data\hwcosim.bsp.

speed to make HCS worthwhile for speed improvements. I'll discuss other uses for HCS where simulation speed isn't the problem.

I haven't covered another option for the HCS connection: Using an Ethernet link, which is even faster. This option is detailed in Xilinx's UG817 Application Note (see Resources). If you are attempting to get the greatest speed, you'll want to use the Ethernet interface instead of JTAG.

INCREMENTAL MADNESS

One of the HCS panel's options is "incremental compile." Normally, your design would be resynthesized every time you relaunch the simulation from ISE. If you are only changing the test bench, this is a huge waste of time, since there are no changes to the hardware design.

"You may wonder why you'd want to use Xilinx ChipScope when you also have the software simulator to plot signals. The most basic reason is that the software simulator cannot plot 'internal' signals running in hardware."

The incremental compile option causes implementation to be skipped, reusing the old bitstream. If you're using the incremental compile option, be sure to disable it when you change your hardware file, otherwise it causes the expected problem of old code being used. This is frustrating when you are sure you fixed a bug, but your fixes don't work because the old bitstream is still being used.

The incremental compile option can also cause odd-sounding error messages and you may still have to occasionally use the "Cleanup Project Files" command under the project menu to clear everything. *C'est la vie*.

On a similar note: You won't get nice synthesis results reports in the project navigator, showing you warnings that may be important. You can run synthesis as part of normal implementation if you want these reports. If you're using Verilog, I also encourage you to use `'default_nettype none` at the start of each file to disable the default behavior that enables implied net creation.

USING CHIPSCOPE WITH HCS

In my previous article, I introduced you to ChipScope, Xilinx's ILA. You can use these tools alongside HCS by sharing the JTAG connection. Within the FPGA, you need to specify that you want the ChipScope core to use a non-default boundary-scan connection, since HCS always uses the default block. The default block connection is via USER1, so instead you need to use another connection (e.g., USER2) for the ChipScope integrated logic controller (ILC). You can select this when inserting the ILC block.

Once you've appropriately configured the device, you can run the simulation with HCS. A few special commands are required to enable ChipScope and HCS to share the JTAG connection.

Command 1: Start iSim, hit the Restart button. Command 2: Start the ChipScope Analyzer and then hit Connect. Command 3: Run the following two commands at the iSim console. Note the `tb/uut` name will vary depending on the device under test—this assumes you've called your testbench `tb` and the HCS module `uut`. Adjust as needed:

```
> scope tb/uut
> hwcosim set shareCable 1
```

Command 4: Set breakpoint on start of testbench code. Command 5: Hit Play in iSim. The bitstream downloads and the testbench starts, but then it breaks on the point you set up. Command 6: Hit Play on the ChipScope Analyzer. You can first test with a "trigger immediate" if you prefer. Command 7: Hit Play on iSim. The code runs as normal, and will trigger ChipScope when you expect.

These are the required instructions to share the Xilinx ChipScope Pro connection with HCS. They may require some alteration. You must ensure the ChipScope ILC core uses USER2 or some other non-default JTAG BSCAN primitive. (Note: A video version is available at ProgrammableLogicInPractice.com.)

You may wonder why you'd want to use ChipScope when you also have the software simulator to plot signals. The most basic reason is that the software simulator cannot plot "internal" signals running in hardware (i.e., you'd need to change the interface to bring them up to the test bench level).

Next I will explain how to add external hardware into the co-simulation. This hardware could run at higher speeds asynchronously to the test bench. Remember the test bench clock typically appears in the hertz or kilohertz range, but now a portion of your hardware design could easily be running in the megahertz range. You need ChipScope to monitor the section of your design running from a different clock to the test bench.

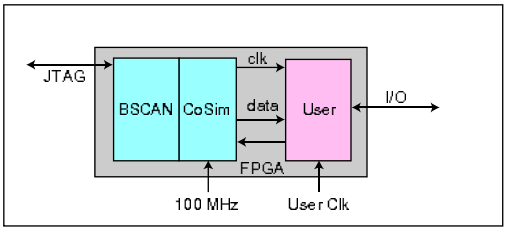


FIGURE 2
Using either JTAG or Ethernet control, the co-simulation module inside the FPGA controls the user module. The clock generated by the co-simulation module runs much slower than the external clock.

CadSoft turns 25!

Celebrate with us and participate in our anniversary quiz*:

or visit: http://www.element14.com/community/community/knode/cadsoft_eagle

25 years CadSoft

1988 - 2013

*Terms and conditions can be viewed here: www.cadsoftusa.com

www.cadsoftusa.com

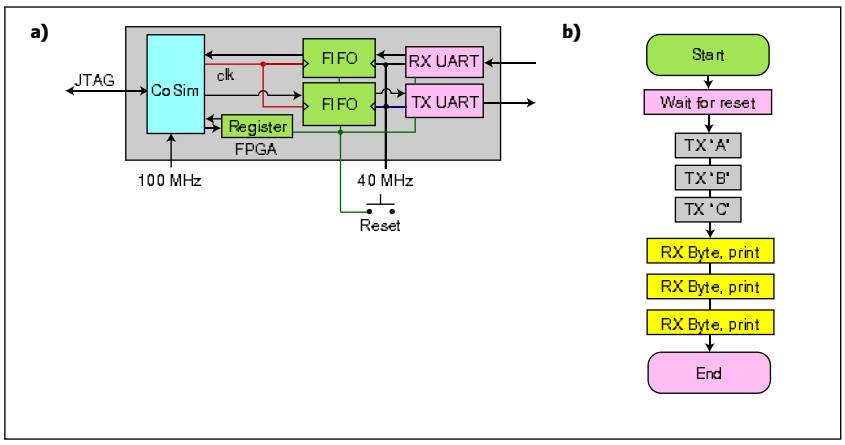


FIGURE 3

a—The serial interface co-simulation's complete setup is shown. FIFOs are used to cross the clock domain between the co-simulation module and the free-running hardware clock. An external reset pin is used to synchronize the test bench to a hardware event. **b—**When running, the serial test bench waits for the user to hit the physical PCB's reset button, sends the string "ABC" out the serial lines, waits for three bytes to be received, and prints the hex value of those bytes.

EXTERNAL HARDWARE CONTROL

Now let's get into something more interesting—interfacing external hardware to the HCS. I'll go through two examples here: interfacing to a serial port and interfacing to DDR memory.

Figure 2 shows the general format. Note that there are suddenly two clock domains: the test bench clock being generated by the HCS controller logic and the clock connected to your hardware module.

Crossing clock domains is always important in FPGA designs, and this is no different. If you are unfamiliar with the usual techniques, see a suitable FPGA reference. I'll use two standard techniques: a first in, first out (FIFO) to pass data between the clock domains and a simple handshaking protocol for some basic status exchange.

When connecting your test bench to the UUT, leave any external ports unconnected. You'll also need a standard constraint (UCF) file, which the simulator will use during implementation to map those "unconnected" ports to the correct physical pins. In ISE, modify the "Simulate Behavioral Model" process properties and add a flag (e.g., -hwcosim_constraints Avt_S6LX9_MicroBoard_UCF_110804.ucf) to the "other compiler options" field.

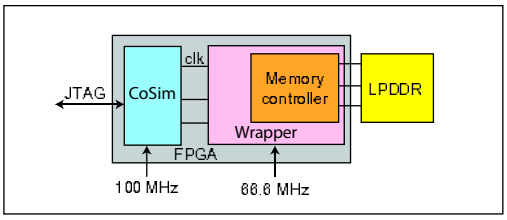


FIGURE 4

Interfacing to the LPDDR chip on the Xilinx Avnet LX9 MicroBoard provides a good example of validating a complex interface with a simple test bench. The wrapper around the Xilinx-provided memory controller contains no logic and only serves to provide the required interface for hardware co-simulation. The computer-based test bench has all the logic to test the DDR chip.

SERIAL KILLER

Reports of the serial port's demise have been greatly exaggerated. I'm going to use it as a simple example of a computer interface. Ultimately, most interfaces end up being a method of shoveling bytes from the computer to your peripheral. In that regard, the serial port isn't so different from any other interface.

Let's look at how you can validate that your computer interface is working perfectly with the real computer. **Figure 3a** shows the test block diagram. **Figure 3b** shows the test's flowchart.

This example shows a few interesting things. First, the test bench running on the computer waits for a signal from the physical hardware to start the test. This signal is the reset signal that is provided by a push button, but you can imagine a variety of other areas where you want the test bench to wait for a hardware event. The test bench uses a simple handshake to acknowledge the reset signal.

Second, a dual-clock FIFO is used to cross the clock domain from the test bench clock to the free-running hardware clock. In reality, remember the test bench clock typically runs much slower than the free-running hardware clock. For certain signals, you may get away with very basic methods to cross the clock domain due to this assumption. However, you should still be careful of the possibility of a metastable event being pushed into your free-running clock domain.

MEMORY TEST

Let's move onto another example. I will demonstrate how to use HCS to test a DDR memory interface. This example is typical of any high-speed interface. Often you want to validate that the hardware works before going deeper into the design/validation process. This means testing the board layout, the soldering of a prototype version, and even just general design principles.

For DDR, the design tools (I used Xilinx ISE) have a memory interface generator (MIG), which gives you a fairly high-level interface to the DDR memory. A tiny wrapper

circuitcellar.com/ccmaterials

RESOURCES

C. O'Flynn, "Using Internal Logic Analyzers for FPGAs," *Circuit Cellar* 279, 2013.

ProgrammableLogicIn Practice.com.

Xilinx, Inc., "ISim Hardware Co-Simulation Tutorial: Accelerating Floating Point FFT Simulation," Application Note UG817, 2012.

——, "ISim Hardware Co-Simulation Tutorial: Interacting with Spartan-6 Memory Controller and On-Board DDR2 Memory," Application Note UG818, 2011.

SOURCES

ISE Design Suite, ISE WebPACK design software, PCI Express IP, Avnet Spartan 6LX9 MicroBoard, Platform Cable USB, ChipScope ILA/ILC, and Spartan-6 FPGA SP601 Evaluation kit
Xilinx, Inc. | www.xilinx.com

ABOUT THE AUTHOR

Colin O'Flynn (coflynn@newae.com) has been building and breaking electronic devices for many years, and is currently completing a PhD at Dalhousie University in Halifax, NS, Canada. His most recent work focuses on embedded security, but he still enjoys everything from FPGA development to hand-soldering his prototype circuits. Some of his work is posted on his website at www.newae.com.

around that generated core can be produced with minimal effort. On top of that, there is a simple test bench that validates the calibration procedure, along with some simple read/write tests.

In this example, you could even use the provided simulation models to first validate your test bench. But, if you are targeting other interfaces, you may not have such a luxury.

Xilinx's Application Note UG818 provides an example of this with DDR2 on the Spartan-6 FPGA SP601 board (see Resources). I've modified it to use the LPDDR on the Avnet LX9 MicroBoard.

Figure 4 shows the test setup's design principles. The entire design is minimal, around 500 lines of Verilog code, most of which is port interfaces. The MIG tools also produce a synthesizable test system, but the point of this example is that you can use HCS to quickly validate any high-speed interface. This is especially helpful if PCB engineers are looking for feedback/validation of their layout before you've had time to invest in the FPGA development.

VALIDATION DONE

Hopefully this article has shown you how HCS can make your life easier. If you want to follow along, visit ProgrammableLogicIn Practice.com for complete source code and project files. Videos also demonstrate the example projects, so even if you don't have the specific boards I'm using here, you can still see what the results should look like.

THE DARKER SIDE

Amplifier Classes From A to H

What is an amplifier and how can you choose the right class type for your application? This article defines amplifiers and describes different classes to help you find the best combination of performance, efficiency, and cost for your design.

By Robert Lacoste (France)

COLUMNS

Welcome back to the Darker Side. In my last article, I explained how to wire a simple bipolar transistor to build an amplifier ("Bipolar Transistor Biasing" *Circuit Cellar* 279). Even if I didn't state it explicitly, these amplifiers were part of the so called Class-A amplifiers.

If you are an audiophile, you know there are plenty of other amplifier classes. They are identified by a long list of letters including the common A, AB, B, and D, but also more exotic letters including G, H, and even T. Their advantages and pitfalls are commonly discussed amongst audio geeks. Try to Google "best audio amplifier class" and you will receive more than 39 million answers.

Of course, amplifiers are used for more than just audio applications. Innovative amplifier designs are also required for RF systems, where efficiency is a key concern. Engineers have worked to develop some additional amplifier classes, namely C, E, and F.

This may seem overwhelming, so I thought it was a good subject for this column. Have a seat, as I try to explain the key differences between all these amplifier families.

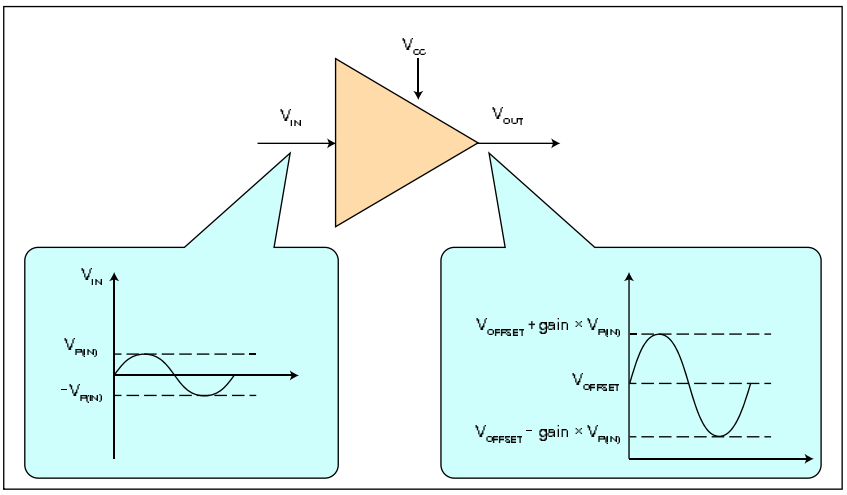


FIGURE 1
An amplifier is a simple concept, but there are plenty of options to implement it.

AMPLIFIER BASICS

Let's start with some basics. For simplicity, I will assume that the amplifier's inputs and outputs are AC coupled through properly sized capacitors, so DC offsets are not a concern. I will also assume that the power supply could be either unipolar (V) or bipolar (±V) and that the signal's polarity is not a problem (if so, it could easily be inverted with another small transistor used as a preamplifier).

What is an amplifier? As shown in **Figure 1**, amplifiers are devices that must accept a given input signal V_{IN}—for example, ranging from $-V_{P(IN)}$ to $V_{P(IN)}$—and generate an amplified version V_{OUT} of the input with some DC offset. So the amplifier's output should be something like:

$$V_{OUT} = V_{OFFSET} + gain \times V_{IN}$$

You may argue that this is more than simplified. You would be correct, as an amplifier can't be restricted to a voltage amplifier. There are concerns regarding input and output impedances, which would lead to the associated current and power gain. Anyway, this article is only about amplifier classes and talking about voltage gain will be enough for this discussion.

Theory is easy, but difficulties arise when you actually want to design a real-world amplifier. What are your particular choices for its final amplifying stage?

CLASS A

The first and simplest solution would be to use a single transistor in linear mode. This

is what I presented in my previous article. **Figure 2** shows a refreshed version.

Basically the transistor must be biased to have a collector voltage close to $V_{CC}/2$ when no signal is applied on the input. This enables the output signal to swing either above or below this quiescent voltage depending on the input voltage polarity.

If you take another look at the schematic, you will understand that a current must continuously flow through the transistor to achieve this biasing set point. This current must never go down to zero, as you want it to stay in the transistor's linear zone.

You could replace the bipolar transistor with some fancy MOSFET or use a pair of transistors to adopt a variant, but the concept would stay the same. It is a Class-A amplifier as long as the transistor is always conducting.

This solution's advantages are numerous: simplicity, no need for a bipolar power supply, and excellent linearity as long as the output voltage doesn't come too close to the power rails. This solution is considered as the perfect reference for audio applications. But there is a serious downside.

Because a continuous current flows through its collector, even without an input signal's presence, this implies poor efficiency. In fact, a basic Class-A amplifier's efficiency is barely more than 30%. It could be increased to 40% if the collector load is replaced with a coupling transformer, but that's still low. And this poor efficiency means heat.

As an example, assume you want to build a stereo 2 × 100-W Class-A amplifier. If you do the math, you'll find that around 300 W would be continuously dissipated in the amplifier's transistors. This would be enough to keep your room at a warm temperature.

CLASS B

How can you improve an amplifier's efficiency? You want to avoid a continuous current flowing in the output transistors as much as possible.

Class-B amplifiers use a pair of complementary transistors in a push-pull configuration (see **Figure 3**). The transistors are biased in such a way that one of the transistors conducts when the input signal is positive and the other conducts when it is negative. Both transistors never conduct at the same time, so there are very few losses. The current always goes to the load. Plenty of variants exist (e.g., with coupling transformers), but a Class-B classification always means each transistor conducts 50% of the time.

A Class-B amplifier has more improved efficiency compared to a Class-A amplifier. This is great, but there is a downside, right?

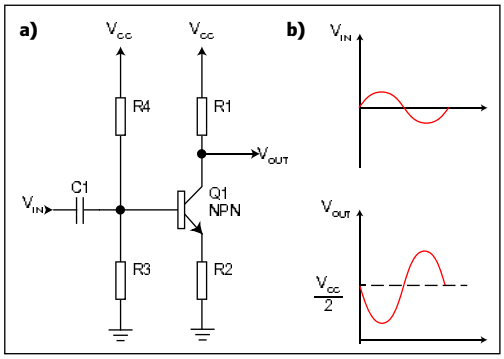


FIGURE 2
a—A Class-A amplifier can be built around a simple transistor. **b**—The transistor must be biased in so it stays in the linear operating region (i.e., the transistor is always conducting).

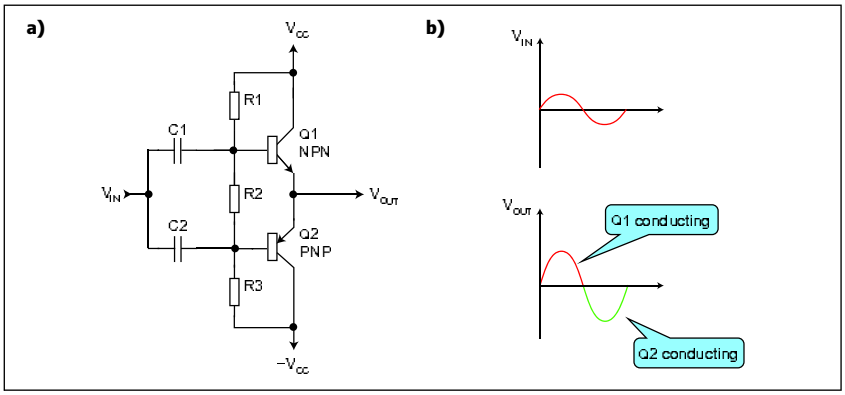


FIGURE 3
a—Class-B amplifiers are usually built around a pair of complementary transistors. **b**—Each transistor conducts 50% of the time. This minimizes power losses, but at the expense of the crossover distortion at each zero crossing.

The answer is unfortunately yes.

The downside is called crossover distortion. The problem is that it is impossible to ensure that one transistor will start to conduct exactly when its counterpart stops, whatever its external conditions (e.g., transistors' temperature, dispersion in transistor's behaviors, input signal frequency, etc.). Therefore, a Class-B amplifier suffers from a small distortion at each signal's zero crossing. It could be small when the amplifier is properly designed, but it will never be as good as a Class-A amplifier.

CLASS AB

As its name indicates, Class-AB amplifiers are midway between Class A and Class B.

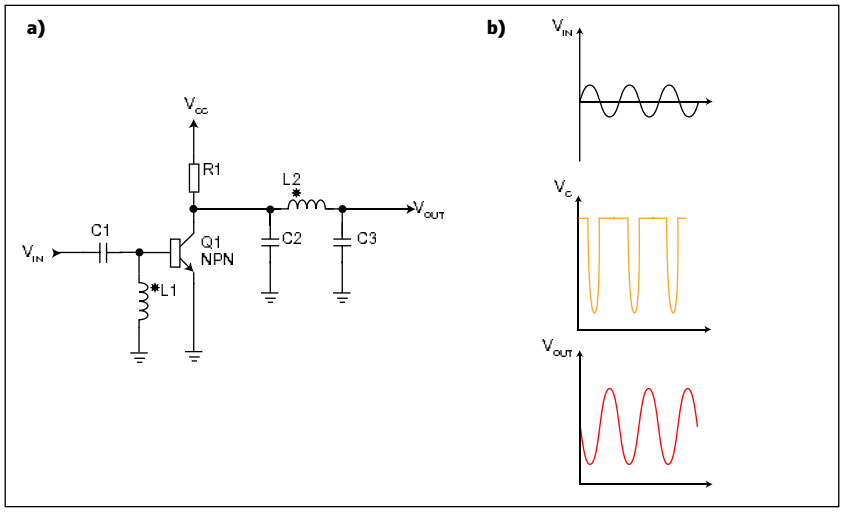


FIGURE 4
a—A Class-C RF amplifier's concept is shown. R1 could be replaced by another inductor to avoid any lossy component except the transistor. **b**—For Class-C operation, the transistor only conducts a small amount of time. The output signal is reconstructed thanks to the output filter.

COLUMNS

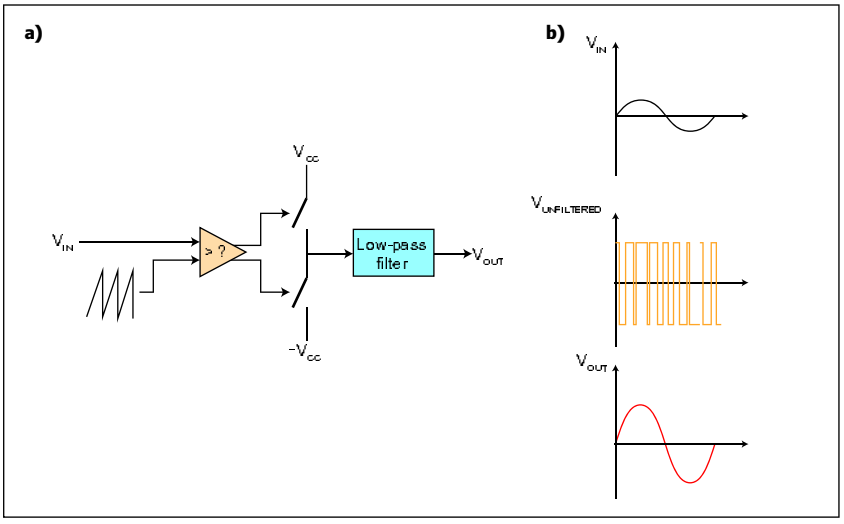


FIGURE 5
a—A Class-D amplifier is a type of digital amplifier. **b—**The comparator's output is a PWM signal, which is amplified by a pair of low-loss digital switches. All the magic happens in the output filter.

Have a look at the Class-B schematic shown in **Figure 3**. If you slightly change the transistor's biasing, it will enable a small current to continuously flow through the transistors when no input is present. This current is not as high as what's needed for a Class-A amplifier. However, this current would ensure that there will be a small overall current, around zero crossing.

Only one transistor conducts when the input signal has a high enough voltage (positive or negative), but both will conduct around 0 V. Therefore, a Class-AB amplifier's efficiency is better than a Class-A amplifier but worse than a Class-B amplifier. Moreover, a Class-AB amplifier's linearity is better than a Class-B amplifier but not as good as a Class-A amplifier.

These characteristics make Class-AB amplifiers a good choice for most low-cost designs. Nearly all low-to-medium-range audio amplifiers used to be Class AB, at least until the proliferation of Class-D amplifiers, as I will explain.

CLASS C

Rather that continuing to talk about audio amplifier with Class D, I found it more fun to follow the alphabet. Now I'll discuss Class-C amplifiers (see **Figure 4**).

There isn't any Class-C audio amplifier. This class is devoted to RF amplifiers. Why? This is because a Class-C amplifier is highly nonlinear. How can it be of any use?

An RF signal is composed of a high-frequency carrier with some modulation. The resulting signal is often quite narrow in terms of frequency range. Moreover, a large class of RF modulations doesn't modify the carrier signal's amplitude.

For example, with a frequency or a phase modulation, the carrier peak-to-peak voltage is always stable. In such a case, it is possible to use a nonlinear amplifier and a simple band-pass filter to recover the signal!

Class-C amplifiers go one step further than Class-B amplifiers. With Class A, the transistor always conducts; with Class B, it conducts half of the time; and with Class C, the transistor conducts only a small percentage of the time (i.e., when the input voltage is close to its peak).

The voltage on the transistor's collector is then sharply dropped one time per period. As illustrated, this signal then has high amplitude and the same frequency as the input signal. However, it is far from a clean sine signal. A proper band-pass or low-pass filter can recover the desired amplified signal.

As you may imagine, designing a properly working Class-C amplifier is a bit more complex than designing with Class A, B, or AB, but the advantage is efficiency.

A Class-C amplifier can have good efficiency as there are no lossy resistors anywhere. It goes up to 60% or even 70%, which is good for high-frequency designs. Moreover, only one transistor is required, which is a key cost reduction when using expensive RF transistors. So there is a high probability that your garage door remote control is equipped with a Class-C RF amplifier.

Join the Elektor Community

Take out a GOLD Membership now!

elektor GOLD membership

Your GOLD Membership contains:

- 8 Regular editions of Elektor magazine in print and digital
- 2 Jumbo editions of Elektor magazine in print and digital (January/February and July/August double issues)
- Elektor annual DVD-ROM
- A minimum of 10% DISCOUNT on all products in Elektor.STORE
- Direct access to Elektor.LABS
- Direct access to Elektor.MAGAZINE; our online archive for members
- Elektor.POST sent to your email account (incl. 25 extra projects per year)
- An Elektor Binder to store these 25 extra projects
- Exclusive GOLD Membership card

ALSO AVAILABLE:

The all-paperless GREEN Membership, which delivers all products and services, including Elektor.MAGAZINE, online only.

elektor GREEN membership

Take out your Membership now at www.elektor.com/member

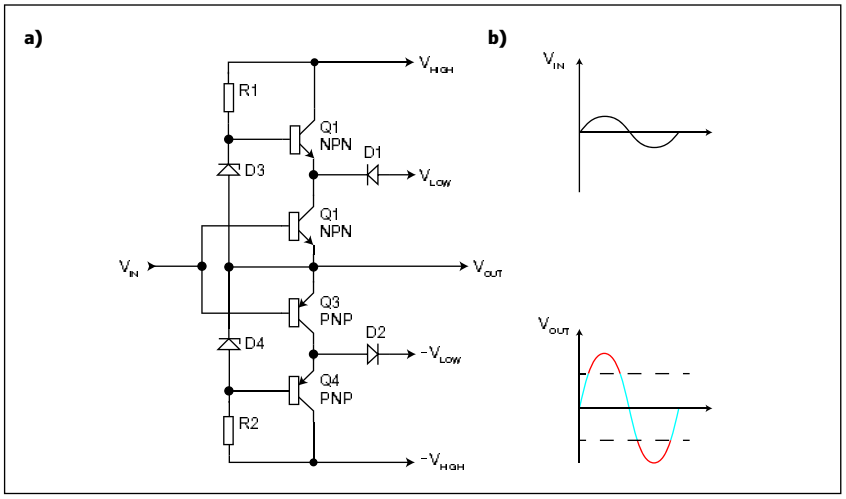


FIGURE 6

a—A Class-G amplifier uses two pairs of power supply rails. **b**—One supply rail is used when the output signal has a low power (blue). The other supply rail enters into action for high powers (red). Distortion could appear at the crossover.

CLASS D

Class D is currently the best solution for any low-cost, high-power, low-frequency amplifier—particularly for audio applications. **Figure 5** shows its simple concept.

First, a PWM encoder is used to convert the input signal from analog to a one-bit digital format. This could be easily accomplished with a sawtooth generator and a voltage comparator as shown on **Figure 5**.

This section's output is a digital signal with a duty cycle proportional to the input's voltage. If the input signal comes from a digital source (e.g., a CD player, a digital radio, a computer audio board, etc.) then there is no need to use an analog signal anywhere. In that case, the PWM signal can be directly generated in the digital domain, avoiding any quality loss.

This PWM signal is then amplified. The beauty of the Class-D architecture is that amplifying a digital signal can be done with high efficiency since the transistors are used as full on/off switches. Power dissipation only occurs during the transitions from one logic state to the other and can be minimized if the transistors are fast enough.

After this amplifying stage, there is a high power signal, but it's still in a PWM format. How can you reconstruct the high-power analog output you were looking for? You can simply use a low-pass filter! Moreover, if you are designing an audio amplifier, then the loudspeaker itself could be part of the filter (it is an inductor), which reduces the bill of material.

As you may have guessed, Class-D amplifiers aren't free from difficulties. First, as for any sampling architecture, the PWM frequency must be significantly higher than the input signal's highest frequency to avoid aliasing. Nyquist sampling theorem says twice higher, but this would require a perfect brick-wall output filter. To keep the filter cost under control, the PWM frequency must be a lot higher, usually 10 times higher than the input signal frequency.

For a 10-Hz-to-20-kHz audio signal, this translates into PWM frequencies in the hundreds of kilohertz, (e.g., 200 kHz). Such a frequency corresponds to a 5-µs period (i.e., 1/200,000).

Now assume you want a 16-bit amplitude accuracy. This implies the position time of each PWM edge must be precise with a 1/65,536 relative accuracy. Therefore, the switching stage's design must not introduce

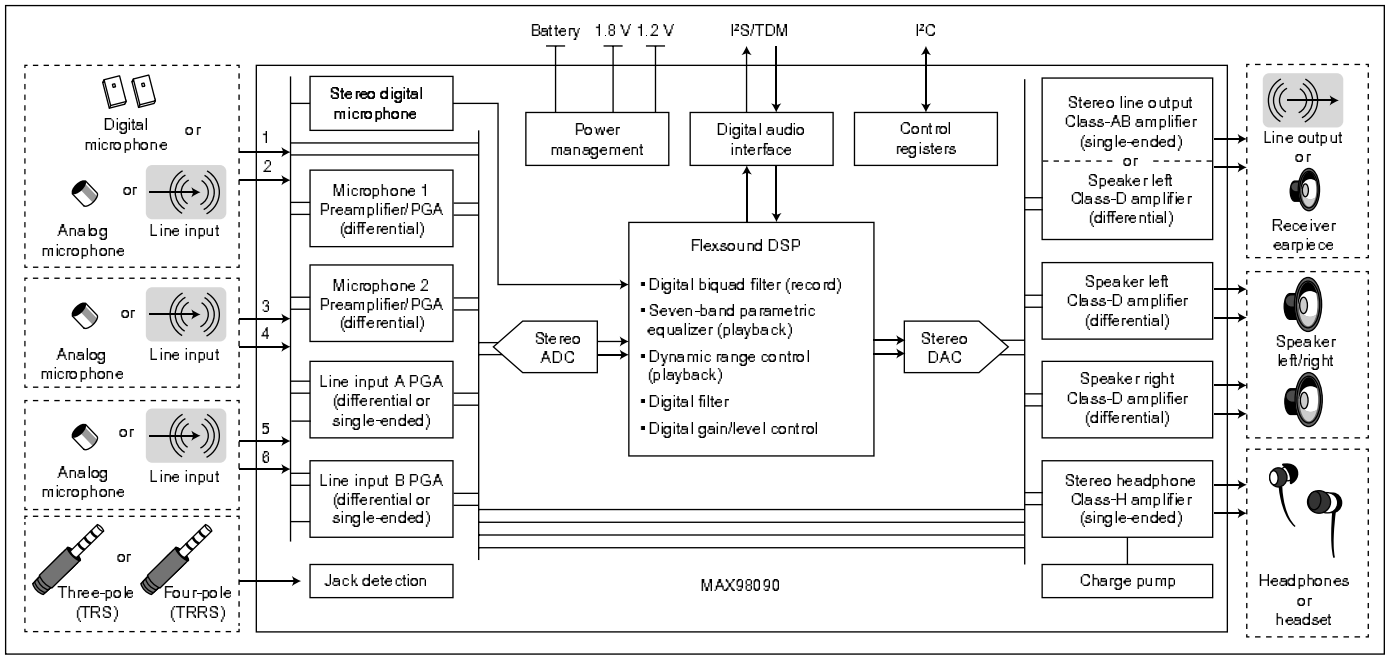


FIGURE 7

Maxim Integrated's MAX98090 ultra-low-power stereo audio codec is a good example of a high-integration multi-mode power amplifier. (Content courtesy of Maxim Integrated)

random variations higher than 76 ps (i.e., 5 µs/65,536) This is quite low, isn't it? This explains why building a good Class-D amplifier is a little more complex than drafting its block diagram.

The second concern with Class-D amplifiers is related to electromagnetic compatibility (EMC). Switching power signals at high speeds is the best way to generate strong electromagnetic fields (EMFs). This is especially true when the output is connected to long unshielded wires such as the ones between an audio amplifier and the loudspeakers.

Class-D amplifier designers must be cautious regarding the output low-pass filter's performance. And they must check the compliance to EMC standards as soon as possible in the design cycle.

Class-D amplifiers are now used everywhere, so solutions do exist. However, their implementation is not as straightforward as it looks. As always, the best solution is to use a pre-integrated solution (e.g., a Class-D IC from one of the many suppliers such as Texas Instruments, Maxim Integrated, and Linear Technology). Then you must look at its reference design and either carefully reproduce it or understand exactly what you are doing before making a modification.

CLASS E AND CLASS F

Remember that Class C is devoted to RF amplifiers, using a transistor conducting only during a part of the signal period and a filter. Class E is an improvement to this scheme, enabling even greater efficiencies up to 80% to 90%. How?

Remember that with a Class-C amplifier, the losses only occur in the output transistor. This is because the other parts are capacitors and inductors, which theoretically do not dissipate any power.

Because power is voltage multiplied by current, the power dissipated in the transistor would be null if either the voltage or the current was null. This is what Class-E amplifiers try to do: ensure that the output transistor never has a simultaneously high voltage across its terminals and a high current going through it.

This may seem impossible, but remember that an RF amplifier is only used for a narrow frequency band around the RF carrier frequency. This enables you to use two tricks to design a Class-E amplifier.

First, as for Class-D amplifiers, the transistor must be used as much as possible as an on/off switching device and not in its linear region (as with Class-C amplifiers). This requires a proper biasing and a very fast transistor.

Second, a specific tuning of the output impedance-matching filter could ensure that the voltage across the transistor is minimal when the transistor switches from one state to the other. Basically the output filter must have the proper complex impedance to introduce the required phase shift between current and voltage.

Going into further detail would require a full article. If you are interested I recommend you read N. O. Sokal's "Class-E RF Power Amplifiers" (*QEX* magazine, 2001).

I have a final word about Class F. This is an improvement to Class E, optimizing the impedance-matching circuits one step further to reduce the effects of harmonics and increase the efficiency a little more. Similar to Class E, Class F is only suitable for narrow-frequency bands, so it is mainly used for RF applications.

CLASS G AND CLASS H

Class G and Class H are quests for improved efficiency over the classic Class-AB amplifier. Both work on the power supply section. The idea is simple. For high-output power, a high-voltage power supply is needed. For low-power, this high voltage implies higher losses in the output stage.

What about reducing the supply voltage when the required output power is low enough? This scheme is clever, especially for audio applications. Most of the time, music requires only a couple of watts even if far more power is needed during the *fortissimo*. I agree this may not be the case for some teenagers' music, but this is the concept.

Class G achieves this improvement by using more than one stable power rail, usually two. **Figure 6** shows you the concept.

As long as the input signal is low enough, the amplifier is a classic Class-AB amplifier and uses a low-voltage power supply. When the signal amplitudes are higher, another couple of transistors go into action, supplied by a higher voltage power.

The detailed design is more complex, in particular to reduce the distortion during the transitions between each mode. The two power supplies must be generated through high-efficiency DC/DC switching converters, as you don't want to reduce the transistors' losses by increasing losses in the power supply section.

Class H goes one step further in the same spirit. Rather than using two or more power supply rails, it uses a continuously variable switching DC power supply. The power supply's voltage is dynamically adjusted to whatever is required for optimal performances depending on the required signal amplitude. This enables a better efficiency; however, the power supply's performances are deeply correlated to the full amplifier's performances.

WRAPPING UP

The list of amplifier classes seems endless. While preparing this article, I even found some references to a so-called "Class-T" audio amplifier. It appears to be a custom architecture developed by Tripath Company—which is now owned by Cirrus Logic—and is somewhat close to Class D. Very little information is available, but at least you known what the "T" stands for.

Amplifier design is an infinite quest for the best balance between performances, efficiency, and cost. An engineer's goal is to find the best mix.

As an example, **Figure 7** shows a block diagram of Maxim's MAX98090 audio amplifier chip. In the same piece of silicon, there is a stereo audio amplifier that could be configured as a 3-W Class-D speaker amplifier, as a Class-AB amplifier for line output, and an independent ultra-low-power stereo Class-H headphone amplifier. This chip also integrates some preamplifiers as well as an ADC, a DAC, and a DSP, but that's another story. It's not too bad for less than $3.

I'm sure you will find your way through a manufacturer's website for even more impressive silicon, but I would like to wrap up this article by clarifying an important point. Designing an amplifier isn't just about selecting which class is the best for your application. This may be enough for simple projects, but as a good engineer, you need to check if a more clever approach may be appropriate.

Do you want some examples? In the RF field, an efficient solution was invented in 1936 at Bell Labs by William H. Doherty. His idea was to associate two amplifiers, one in charge of the RF carrier amplification and one providing some extra power when the input signal is strong. Each section could then be independently optimized for best

circuitcellar.com/ccmaterials

RESOURCES

S. C. Cripps, *RF Power Amplifiers for Wireless Communications*, 2nd Edition, Artech House, 2006.

Devialet, "D-Premier White Paper."

E. Gaalaas, "Class-D Audio Amplifiers: What, Why, and How," Analog Devices, Inc., 2006.

J. Honda and J. Adams, "Class-D Audio Amplifier Basics," International Rectifier, Application Note AN-1071, 2005.

R. Lacoste, "Bipolar Transistor Biasing" *Circuit Cellar* 279, 2013.

Linear Technology, Corp., www.linear.com.

K. Mcknight, "Doherty Amplifier Design," Microwave Monolithic Integrated Circuit (MMIC) Course, Johns Hopkins University, 2009.

D. Self, *Audio Power Amplifier Design Handbook*, 5th Edition, Focal Press, 2009.

N. O. Sokal, "Class-E RF Power Amplifiers," *QEX* magazine, 2001.

H. Theiler, H. Lenhard, and H. Gether, "Class-G/H Amplifiers: How Well Do They Deliver on Their Promise of High Audio Quality and Low Power Consumption?" ams AG.

Wikipedia, "Amplifier."

———, "Class-T amplifier."

Yamaha Corp., "Yamaha Power Amplifier White Paper," 2008.

SOURCES

D-Premier amplifier and ADH technology
Devialet | http://en.devialet.com

MAX98090 Ultra-low-power stereo audio codec
Maxim Integrated, Inc. | www.maximintegrated.com

ABOUT THE AUTHOR

Robert Lacoste lives in France, near Paris. He has 24 years of experience in embedded systems, analog designs, and wireless telecommunications. A prize winner in more than 15 international design contests, in 2003 he started his consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. His book (*Robert Lacoste's The Darker Side*) was published by Elsevier/Newnes in 2009. You can reach him at rlacoste@alciom.com if you don't forget to put "darker side" in the subject line to bypass spam filters.

performances (e.g., with a Class-AB amplifier for the carrier and a Class-C amplifier for the peaks). This so-called "Doherty amplifier" is still currently in use.

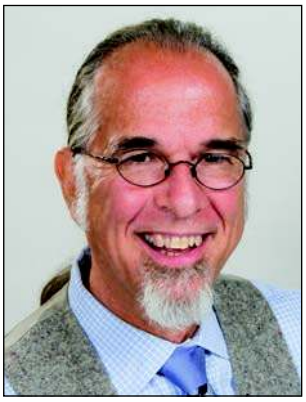
Here are a couple more audio field examples. Yamaha's "Yamaha Power Amplifier White Paper" explains how the company merged a Class-AB and a Class-D amplifier for best cost/performance ratio. Basically, a Class-D amplifier provides a coarse output at high efficiency and is improved by a Class-AB final stage providing fine-tuning at low loss. To me, it seems close to a Class-H amplifier.

In the same spirit, but a more higher-end version, look at Devialet's award-winning D-Premier amplifiers. Its patented ADH technology (which stands for analog digital hybrid) combines a Class-A amplifier used as a reference and eight Class-D digital amplifiers per channel, achieving a total harmonic distortion below 0.001% and an amazing 130 dB of signal-to-noise ratio, which is more than impressive with 240 W of output power!

In such a short article, my goal was not to help you to design a amplifier. I just wanted to give you some information about how to select an adequate topology for your application.

Amplifier classes have been a hot topic since the early days of electronics, so there are plenty of publications and books on this subject. The information available in the Resources could be your starting point.

FROM THE BENCH



Unleash Your Android Device's Potential

Future Technology Devices International's new interface chip turns your Android device into an effective application tool. The chip can be added to any circuit, enabling the Android device to serve as a user I/O device.

By Jeff Bachiochi (USA)

COLUMNS

I like to make or purchase little widgets using new technology. I have an initial desire to learn about a new device before using it in a project. Having a small circuit with an interface to the supported synchronous or asynchronous serial bus makes it compatible with most microcontrollers, so it is easy to work with. If you don't mind spending a little time writing some code on any small microcontroller-based board you may have around, you can exercise the new device.

Sometimes a manufacturer will offer a small circuit to demonstrate the new technology. Many times this is priced low (at cost or even less) because manufacturers hope if you play with the circuit, you will use it in a future design. As a widget, the interface bus enables you to connect it to any microcontroller-based board, such as the Arduino.

Arduino has library routines that enable me to communicate with any serial protocol and incorporate an LCD and maybe some push buttons. It could be a fun project. It would be nice to have this type of tool to use on widgets.

Future Technology Devices International (FTDI) makes a tool that enables you to use an Android device as a user interface to the various protocols. You may be familiar with FTDI, as it's been around for more than 20 years.

FTDI may be most noted as specialists in converting legacy peripherals to USB. Its most popular product presents a USB interface for direct connection to a microcontroller's TTL UART pins. You can replace what were the normal RS-232 converter circuitry and a DB9/25 (i.e., legacy serial port) with one of these devices and a USB connector, to make your microcontroller USB compatible. You don't need to know anything about the intricacies of USB. It's a great idea and it's easy to implement.

A few years ago, FTDI announced a commitment to supporting the Android Open Accessories initiative. It wanted to enable engineers to make use of tablets and smartphones utilizing the Android OS. This led to last year's introduction of the FT311D multi-interface Android host IC. This device provides an instant bridge from an Android USB port(B) to peripheral hardware over general-purpose input/output (GPIO), UART, PWM, I²C Master, SPI Slave, or SPI Master interfaces. Let's take a look at how FTDI's apps and the FT311D can be used to make a worthy addition to your toolbox.

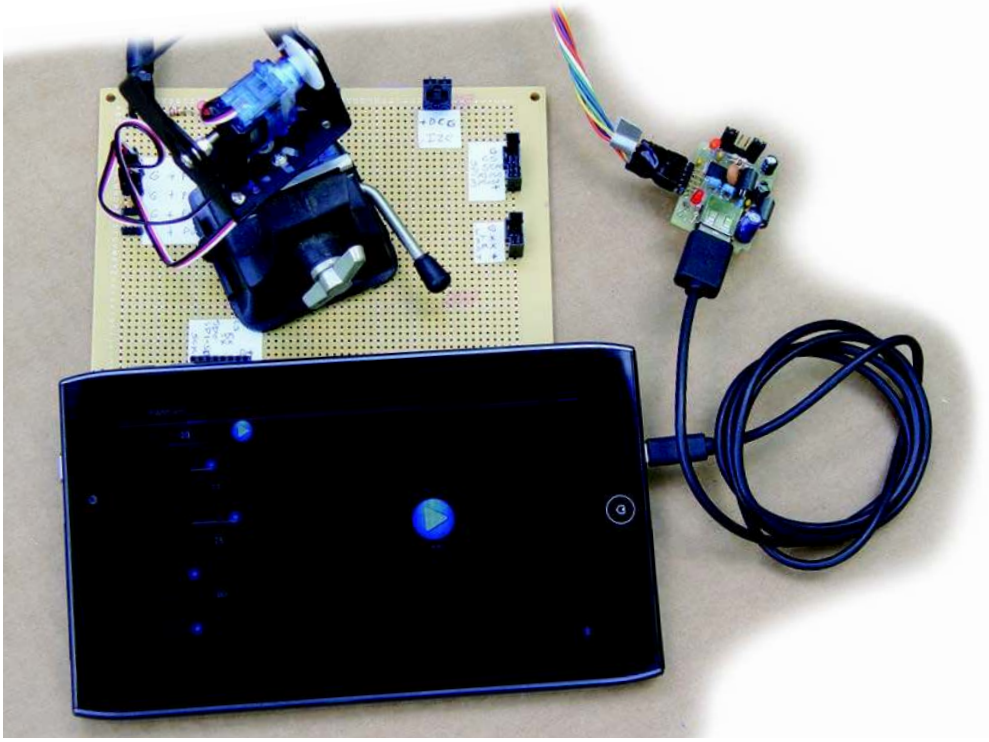


PHOTO 1
I used a couple of DC servomotors to test the PWM functions; however, this interface is best for driving LED brightness or as motor controller inputs.

USB ANDROID HOST IC

The FT311D is a full-speed USB host targeted at providing access to peripheral hardware from a USB port on an Android device. While an Android device can be a USB host, many are mobile devices with limited power. For now, these On-The-Go (OTG) ports will be USB devices only (i.e., they can only connect to a USB host as a USB device).

Since the USB host is responsible for supplying power to a USB peripheral device, it would be bad design practice to enable a USB peripheral to drain an Android mobile device's energy. Consequently, the FT311D takes on the task of USB host, eliminating any draw on the Android device's battery.

All Android devices from V3.1 (Honeycomb) support the Android Open Accessory Mode (AOAM). The AOAM is the complete reverse of the conventional USB interconnect. This game-changing approach to attaching peripherals enables three key advantages. First, there is no need to develop special drivers for the hardware; second, it is unnecessary to root devices to alter permissions for loading drivers; and third, the peripheral provides the power to use the port, which ensures the mobile device battery is not quickly drained by the external hardware being attached.

Since the FT311D handles the entire USB host protocol, USB-specific firmware programming isn't required. As the host, the FT311D must inquire whether the connected device supports the AOAM. If so, it will operate as an Open Accessory Mode device with one USB BULK IN endpoint and one USB BULK OUT endpoint (as well as the control endpoint.) This interface will be a full-speed (12-Mbps) USB enabling data transfer in and out.

The AOAM USB host has a set of string descriptors the Android OS is capable of reading. These strings are (user) associated with an Android OS application. The Android then uses these strings to automatically start the application when the hardware is connected. The FT311D is configured for one of its multiple interfaces via configuration inputs at power-up. Each configuration will supply the Android device with a unique set of string descriptors, therefore enabling different applications to run, depending on its setup.

The FT311D's configuration determines whether each application will have access to several user interface APIs that are specific to each configuration. Next, let's take a look at these interfaces in detail.

HOST INTERFACES

The GPIO interface consists of seven signals the user can control (see **Table 1**). There are four GPIO commands; each uses a four-byte command format: `ConfigPort`, `ReadPort`, `WritePort`, and `ResetPort` (see **Table 2**). `ConfigPort` enables the application to define each GPIO bit's function. `ReadPort` is sent to the application any time an input changes state on the interface. `WritePort` enables the application to change any output bit on the interface. Should the GPIO interface need to be reset and configured as all inputs,

Signal Name	FT311D Pin Number	I/O Type	Description
GPIO(0)	23	IO	GPIO data bit 0, bidirectional
GPIO(1)	24	IO	GPIO data bit 1, bidirectional
GPIO(2)	25	IO	GPIO data bit 2, bidirectional
GPIO(3)	26	IO	GPIO data bit 3, bidirectional
GPIO(4)	29	IO	GPIO data bit 4, bidirectional
GPIO(5)	30	IO	GPIO data bit 5, bidirectional
GPIO(6)	31	IO	GPIO data bit 6, bidirectional

TABLE 1
In GPIO mode, any of the seven signals can be an input or output bit.

Command	Byte 1	Byte 2	Byte 3	Byte 4
`ConfigPort`	0x11	0x00	Bits set to 1 are considered outputs	Bits set to 1 are considered inputs
`ReadPort`	0x12	GPIO Status	0x00	0x00
`WritePort`	0x13	GPIO Data	0x00	0x00
`ResetPort`	0x14	0x00	0x00	0x00

TABLE 2
This is the command byte structure of the available GPIO commands.

Signal Name	FT311D Pin Number	I/O Type	Description
PWM(0)	23	O	PWM Channel 0
PWM(1)	24	O	PWM Channel 1
PWM(2)	25	O	PWM Channel 2
PWM(3)	26	O	PWM Channel 3

TABLE 3
In PWM mode, four signals can produce a PWM output.

Command	Byte 1	Byte 2	Byte 3	Byte 4
`SetPeriod`	0x21	0x00	0x00	Period in milliseconds for all channels (1–255)
`SetDutyCycle`	0x22	PWM Channel	0x00	Percent ON (5–95)
`Reset`	0x23	0x00	0x00	0x00

TABLE 4
This is the command byte structure of the available PWM commands.

Signal Name	FT311D Pin Number	I/O Type	Description
I2C_CLK	23	O	I²C Clock
I2C_Data	24	IO	I²C Data

TABLE 5
Two signals are used in I²C mode.

the application may use the `ResetPort` command.

The PWM interface consists of four PWM output signals, which have period and duty cycle parameters (see **Table 3**). There are three PWM commands; each uses a four-byte command format: `SetPeriod`, `SetDutyCycle`, and `Reset` (see **Table 4**). `SetPeriod` enables the application to define each GPIO bit's function. `SetDutyCycle` is sent to the application any time an input changes state on the interface. Should the GPIO interface need to be reset and configured as all inputs, the application may use the `Reset` command.

The I²C interface uses two signals, clock and data (see **Table 5**). There are four I²C commands; each uses a multi-byte command format: `SetFrequency`, `WritePort`, `ReadPort`, and `Reset` (see **Table 6**). The application used `SetFrequency` to define the I²C clock output's byte timing. `WritePort` enables the application to write a number of data bytes to the I²C Slave device. Note the `WritePort` command is also returned to inform the application of the data transfer's success. `ReadPort` enables the application to request a number of data bytes from the I²C Slave device. Note the `ReadPort` command is also returned to inform the application of the data transfer's success and to pass the data back to the application. Should the I²C interface need to be reset and configured as all inputs, the application may use the `Reset` command.

The UART interface consists of five signals (see **Table 7**). Here there is just one command, `SetConfig`, which uses an eight-byte command format (see **Table 8**). `SetConfig` enables the application to define the UART's configuration and must be sent before any data is transferred. Once configured, the application is free to send and/or receive blocks of up to 256 bytes of data. For light loads, handshaking unnecessary. To move a large amount of data, the RTS and CTS lines should be used.

The last two interfaces are SPI Master and SPI Slave. While the interfaces use the same four signals, they are configured differently depending on whether you want to emulate a master or a slave device. **Table 9** shows how

TABLE 6
This is the command byte structure of the available I²C commands.

Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte N
`SetFrequency`	0x31	Frequency in kilohertz (23, 44, 60 and 92)	0x00	0x00	0x00
`WriteData`	0x32	7-bit address	Options	N	Data bytes (up to 253)
`WriteData`	0x32	7-bit address	Status	N	
`ReadData`	0x33	7-bit address	Options	N	
`ReadData`	0x33	7-bit address	Status	N	Data bytes (up to 253)
`Reset`	0x34	0x00	0x00	0x00	0x00

TABLE 7
Five signals are used in UART mode.

Signal Name	FT311D Pin Number	I/O Type	Description
UART_TXD	23	O	Data transmitter
UART_RXD	24	I	Data receiver
UART_RTS#	25	O	Ready to send (handshake)
UART_CTS#	26	I	Clear to send (handshake)
UART_TX_ACTIVE	29	O	TX Enable (RS-485)

TABLE 8
This is the command byte structure of the available UART commands.

Command	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
`SetConfig`	Baud rate, in little-Endian format				Data bits (7 or 8)	Stop bits (1 or 2)	Parity (0 = N, 1 = O, 2 = E, 3 = M, 4 = S)	Flow control (0 = N, 1 = RTS/CTS)
`SendData`	N (1–256) bytes of data							
`ReadData`	N (1–256) bytes of data							

Signal Name	FT311D Pin Number	I/O Type	Description
SPI_S-SS#	26	I	SPI Select input
SPI_S_CLK	29	I	SPI Clock input
SPI_S_MOSI	30	I	SPI Slave data input
SPI_S_MISO	31	O	SPI Slave data output

TABLE 9
In SPI Slave mode, four signals comprise the interface.

Command	Byte 1	Byte 2	Byte 3
SetConfig	0x51	Mode (0–3)	Bit0 = 0 (MSB first) Bit0 = 1 (LSB first)
SendData	0x52	Send N bytes of data (1–255)	
SendData	0x52	N bytes sent	
ReadData	0x53	Retrieve N bytes of data (1–255)	
Reset	0x54		

TABLE 10
This is the command byte structure of the available SPI Slave commands.

Signal Name	FT311D Pin Number	I/O Type	Description
SPI_M_SS#	26	O	SPI Select output
SPI_M_CLK	29	O	SPI Clock output
SPI_M_MOSI	30	O	SPI Master data output
SPI_M_MISO	31	I	SPI Master data input

TABLE 11
Four signals comprise the interface in SPI Master mode.

the signals are defined in the Slave mode. There are four SPI Slave commands; each uses a multibyte command format: SetConfig, SendData, ReadData, and Reset (see **Table 10**). This SPI Slave device has an array of up to 256 data bytes that can be read or written by a master on the bus. SetConfig enables the application to define the SPI Slave peripheral's configuration. SendData enables the application to write data into the array on the Slave interface. Note, if the SendData command is executed, the two-byte SendData command is sent back to the application as verification. ReadData enables the application to retrieve data from the array on the Slave interface. Should the GPIO interface need to be reset and configured as all inputs, the application may use the Reset command.

Table 11 shows how the signals are defined in the Master mode. There are four SPI Slave commands; each uses a multibyte command format: SetConfig, SendData, ReadData, and Reset (see **Table 12**). This SPI Master device can send or retrieve up to 255 data bytes from a Slave device on the bus. An array of this data is held in the application. SetConfig enables the application to define the SPI Master peripheral's configuration. SendData enables the application to write N data bytes from an array to a Slave interface. Note once the SendData command is executed, a SendData command is returned with the number of bytes actually received. ReadData enables the application to retrieve N data bytes from a Slave interface. Should the GPIO interface need to be reset and configured as all inputs, the application may use the Reset command.

Each interface's format has its own quirks, but is kept as simple as possible. I won't discuss writing applications because FTDI has demo applications for each interface (see Resources). Since you only need rudimentary functions to enable you to exercise the FT311D, you can use these applications directly in support of

Command	Byte 1	Byte 2	Byte 3	Bytes 4–7
SetConfig	0x61	Mode (0–3)	Bit0 = 0 (MSB first) Bit0 = 1 (LSB first)	ClockSpeed, in little-Endian format (up to 24 MHz)
SendData	0x62	Send N bytes of data (1–255)		
SendData	0x62	N bytes of data sent		
ReadData	0x63	Retrieve N bytes of data (1–255)		
ReadData	0x63	N bytes of data retrieved		
Reset	0x64			

TABLE 12
This is the command byte structure of the available SPI Master commands.

this tool, which will enable you to investigate new technologies that come with one of these simple interfaces. Let's look at the circuit required for this project.

PROJECT HARDWARE

A USB host must supply 5 V to the Android device. While the FT311D runs on 3.3 V, it does have 5-V tolerant inputs. I started with a 2.1-mm power jack suitable for most 5-V wall warts. A 78L33 regulator will supply 100 mA; however, the FT311D's required operating current is only 25 mA. Two LEDs are used for status. LED1 is steadily illuminated when the USB is connected and functional. If connected but nonfunctional, it will blink an error code. LED2 indicates the circuit is powered. An external 12-MHz crystal or resonator is required. This feeds an internal PLL to enable internal execution at 48 MHz (which is needed for USB 2.0).

Figure 1 shows the circuit diagram. Three internally pulled-up configuration inputs enable selection of one of six operation modes. The input leaves various combinations of these pins open or applies a jumper to short the pins to ground. Finally, a single-keyed connector provides access to all bus signals plus 5 V, 3V3, and ground. If you want to use this tool with an external circuit that already has 5 V, you may power it from that circuit through this 10-pin connector.

I found some slick-looking enclosures (blue translucent) that are small and inexpensive (less than $3) for this circuit. A nice enclosure always legitimizes a project, giving it a finished appearance. Labeling identifies the jumper configurations and the signal connections.

PWM DEMO

Now I'll discuss the PWM outputs and drive RC servomotors. Each servomotor requires a 1-ms pulse to send the arm, wheel, or horn to the right stop. A 1.5-ms pulse centers it and 2-ms pulse sends it to the left stop. Repeating this pulse takes approximately 40 ms. This delay is not as critical as the other parts of the timing signal. It is essentially the dead time between control signals. If you repeat

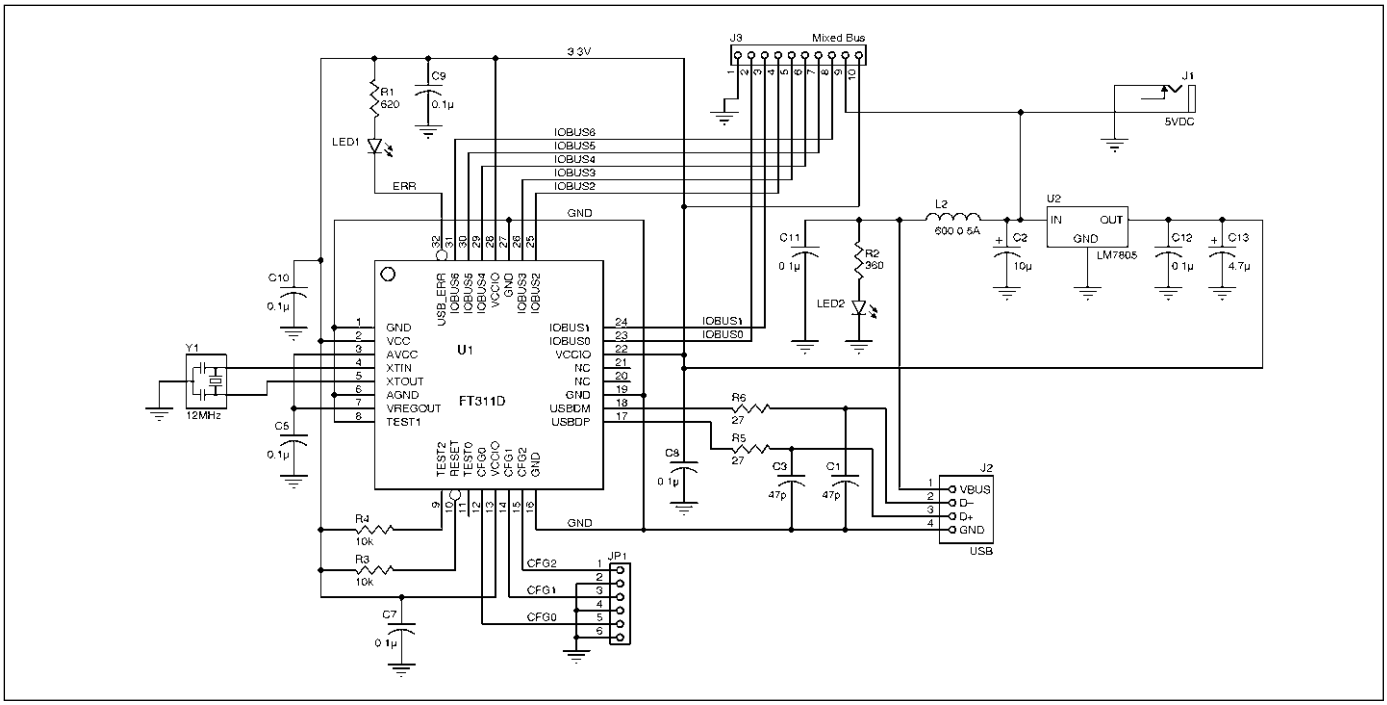


FIGURE 1
The Future Technology Devices International FT311D IC is available in 32-pin LQFP and QFN packages. You can use a small-tip iron (and a little magnification) to hand solder the LQFP version. I used through-hole peripheral components for the remainder of the parts. A keyed 10-pin connector ensures there aren't any connection errors.

the control pulse too quickly (i.e., 10 ms) the servomotor will buzz and jitter. If you repeat the control signals too slowly (i.e., 70 ms) the servomotor will shut off between signals and its position will not remain constant.

With the Servo app, the repetition rate can be controlled for all PWM outputs between 1 and 250 ms (see **Photo 1**). Each pulse output is 5% to 95% of the repetition rate. This is a problem for servomotors. If you wanted to set a minimum 1-ms pulse, you'd need a 20-ms repetition rate so that 5% would equal 1 ms (20 ms × 0.05 ms). At 95%, the pulse would be 10 ms (no more than 2 ms is needed). So you will need to limit the pulse to 10% or use a lower repetition rate. I found that a 4-ms repetition rate worked well with my servomotors without an annoying buzz. However, the PWM API is not designed to run servomotors.

The outputs can source (or sink) up to 8 mA to control LED brightness or motor speed. Many motor drivers operate with PWM inputs such as the old favorite Texas Instruments (TI) L923D quadruple half-H drivers or TI's newer DRV8835. Two modes can often be used with these drivers: the PWM and direction control and the PWM and enable control. The first uses the PWM to determine the 0% to 100% speed. The direction control selects the motor's direction. The second arrangement uses the PWM to determine speed and direction with 50% = OFF. The speed increases in opposite directions depending on whether the PWM is less than or greater than 50% (0% = full ON reverse and 100% = full ON forward).

I²C DEMO

The serial communication protocols are the most widely used in the industry, as they require a low number of I/O lines. On the synchronous side (i.e., those providing a clock), the I²C uses only two signals, clock and data. The open collector bus style enables any connected device to pull either bus line low. While this can hang communications, when properly used it enables a slow peripheral to halt further clocking until it becomes ready to respond.

The Bus Master supplies the clock to other devices. The data line can change only while the clock line is low. The data line changing while the clock line is high indicates a formatted transmission's beginning (start bit) or end (stop bit).

The format consists of multiple 9-bit transmissions with the first 8 bits indicating a desired device's 7-bit address and a bit indicating whether the transmission wants to read or write to a device. The ninth bit is an opportunity for the addressed device to say, "Got it, continue." Additional bytes contain 8-bit data.

Each device has its own factory set address (or address range). Data bytes will be read from or written to the device's registers, always starting with the first register. Many devices use a register pointer to select one of these registers, which eliminates unnecessary data transfers. In this case, the first data byte written will change this pointer. The next data bytes read (or written) will go the pointed to register. This pointer is auto-incrementing.

Photo 2 shows the demo, which enables the user to set the I²C clock's frequency value. I've attached a past project that exposes the compass chip's Honeywell HMC5843 I²C bus. This device uses a register pointer with 13 registers. Field strength measurements of the X, Y, and Z axes are available in a 12-bit signed format. When using a registered device, the first data byte written goes to the pointer register. Reads must be preceded by a write (to properly set the pointer). The number of actual data bytes written will be N-1 (as the pointer is written using the first byte).

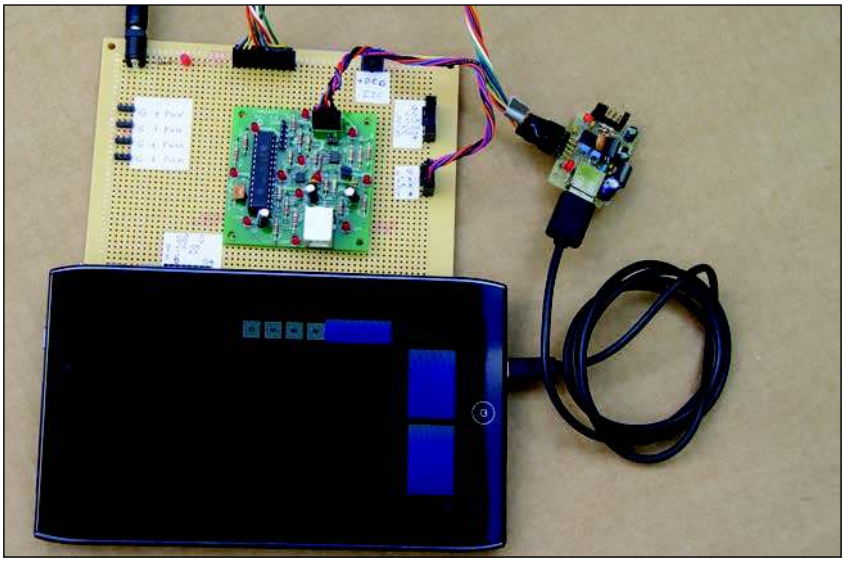


PHOTO 2
I used a Honeywell HMC5843 digital compass IC for the demo. Three-axis magnetic measurements are available as signed integers.

SPI MASTER DEMO

SPI adds an extra data line to an I²C bus's clock and data lines. With two data lines, the data direction is fixed on each. This enables data to simultaneously move in both directions (potentially twice as fast as I²C). Using fixed data directions also enables the bus to be TTL-compatible (as opposed to open-collector). Since

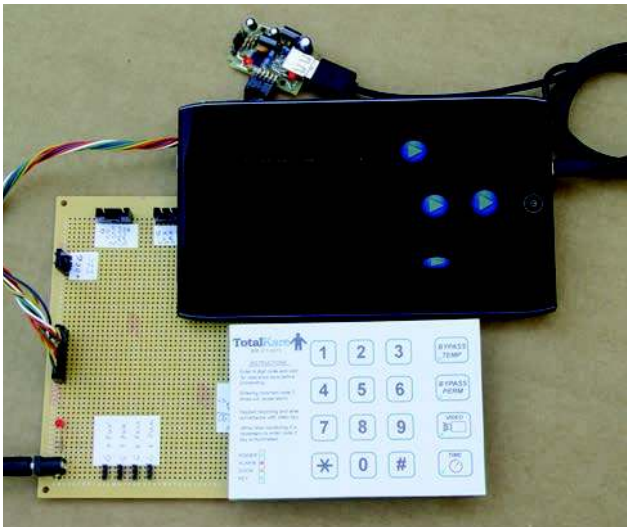


PHOTO 3
I used the SPI Master demo to read key presses and set LED states on this SPI slave 16-key touch panel.

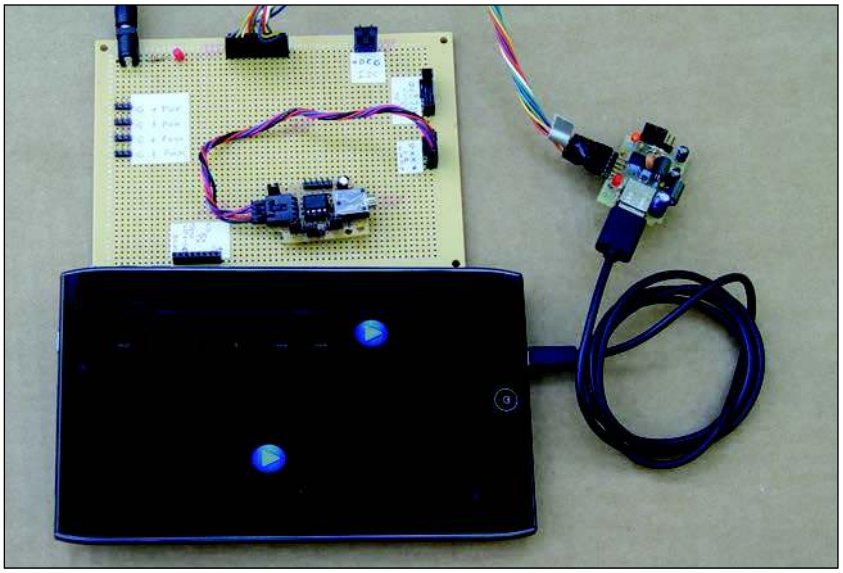


Photo 4
Many of my projects use printable ASCII text commands and replies. This enables a serial terminal to become a handy user I/O device. This current probe circuit outputs its measurements in ASCII-printable text.

SPI devices don't have fixed addresses, a chip select (separate enable) is required for each slave device to exist on the same bus.

There is no standard SPI format. Therefore it is necessary to know the format of the device with which you are working. SPI devices pass information in some fixed-bit packet, usually some factor of 8 (since the hardware is an 8-bit shift register). For instance, using an 8-bit format, the first byte sent may be a command with 1-bit R/W + 7-bit pointer format. A 0x00 may mean set the pointer to b'0000000' and set up to write any following bytes. A 0x80 may mean set the pointer to b'0000000' and set up to read any additional bytes.

Note that while this byte is being clocked into the Slave device, the Slave device is clocking something out to the Master. Since a Slave device can't determine how to answer a question before is was asked, this byte is useless (although it may actually hold data from the last request's next register). Unlike I²C, a Slave device cannot hold the clock line low, requesting a wait while it prepares to reply to the command with the right data, so a dummy byte is often added to the format to give the Slave device some processing time. This means when the command is to read data, dummy data bytes are received while the complete command (and dummy byte are sent), then additional dummy bytes are sent to enable the data bytes you want to receive to be clocked out.

A SPI touch keypad is attached as a Slave device to the SPI Master application (see **Photo 3**). This keypad has four commands: read LED status, read keypad status, read keypad key, and write LED state. All read commands are three bytes: the command, a dummy byte (to give the Slave device time to prepare a reply), and a dummy byte to clock in the (one-byte) reply. The write command is two bytes: the command and the data byte (no data needs be read back).

UART DEMO

The UART demo is probably the easiest mode to use. Like the I²C mode, it requires just two lines: TX and RX. These are TTL-level signals, not the inverted RS-232 levels associated with an asynchronous serial port link. This means all data idles at logic 1.

Asynchronous means the clock is not transmitted along with the data. While the format must be predetermined, each 10-bit packet's starting edge is used to sync the data with the predetermined clock. The packet format consists of a start bit + the number of data bits + any parity bit + the number of stop bits (that's 10 bits using the 8-none-1-bit format). The clock speed or baud rate has a broad number of standard speeds.

Photo 4 shows how I attached a current probe project to the UART application. This circuit continually measures the current that passes through an attached probe. All commands and replies for this project are in ASCII-formatted text. Three commands are used: `RC` (read current), `RP` (read probe type), and `WP` = (value) (write probe type). `RC` returns the present measured value in milliamperes as C = (value) mA. `RP` returns P = (value). The write probe type command also returns P = (value). The probe type is the maximum amps for which the probe is designed and changes the measurement values' scaling.

TIED DOWN OR UNLIMITED POTENTIAL RELEASED?

Multimeters are great tools. They have portability that enables them to be brought to wherever a measurement must be made. An Android device has this same ability. Since applications can be written for these devices,

circuitcellar.com/ccmaterials

RESOURCES

Arduino, http://arduino.cc.

Future Technology Devices International, Ltd., "FT31xD Demo APK User Guide," Application Note 208, 2013.

———, "Software Application Development FT31xD Android Programmers Guide," Application Note FT_000532, 2013.

SOURCES

FT311D USB Android host IC
Future Technology Devices International, Ltd. | www.ftdichip.com

HMC5843 Digital compass IC
Honeywell International, Inc. | www.honeywell.com

L923D Quadruple half-H driver and DRV8835 motor driver
Texas Instruments, Inc. | www.ti.com

ABOUT THE AUTHOR

Jeff Bachiochi (pronounced BAH-key-AH-key) has been writing for *Circuit Cellar* since 1988. His background includes product design and manufacturing. You can reach him at jeff.bachiochi@imaginethatnow.com or at www.imaginethatnow.com.

they make a great portable application tool. Until the AOAM's release, there was no way for these devices to be connected to any external circuitry and used as effective tool.

I think FTDI has bridged this gap nicely. It provided a great interface chip that can be added to any circuit that will enable an Android device to serve as an effective user I/O device. I've used the chip to quickly interface with some technology to discover its potential or just test its abilities. But I'm sure you are already thinking about the other potential uses for this connection.

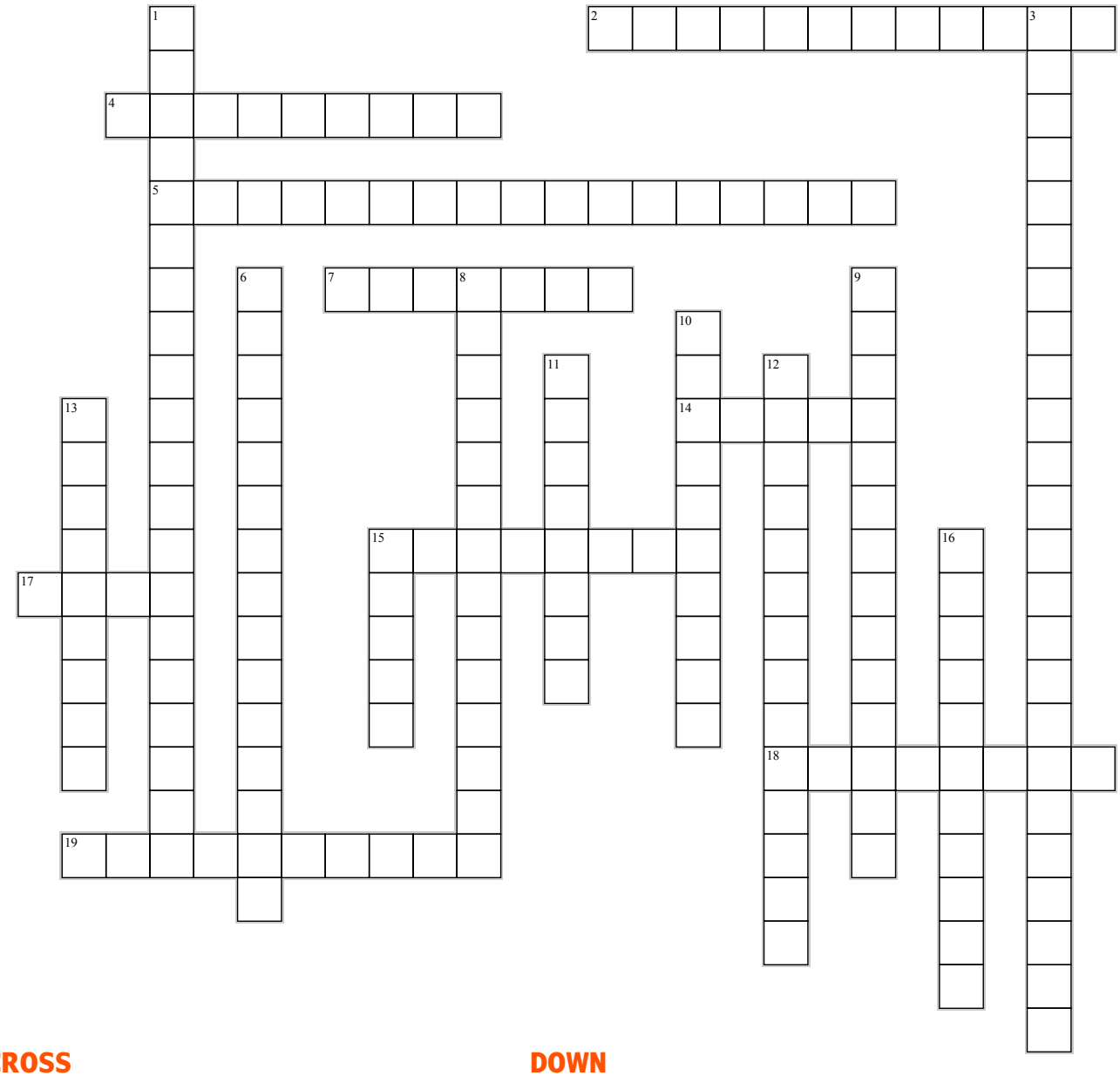
One note about the demos provided. Other than the GPIO and PWM demos, all data is displayed as ASCII text. For anyone who works with data, you know that you must work in a format in which all values can be entered and displayed, and this is not printable ASCII text. The demo apps released by FTDI only allow ASCII characters, which severely limits their usefulness. I know the source code is available for every demo and I can make improvements and recompile, but I had to wonder. When I asked FTDI about this, the company responded with: "No one has ever asked for that. I'll put in a request." Please note this isn't a device issue, but merely a shortsighted design issue of the demo applications.

I have a final comment about Android applications in general. If you think the AOAM has future potential, but you want to know what's involved with writing Android applications for a specific purpose, send me an e-mail and I'll add this to my list of future projects!

Author's Note: As of this printing, FTDI has released all of the demos, providing users with a choice of ASCII, Hex, or Decimal characters for entry/display. Now that is quick customer service!

CROSSWORD DECEMBER 2013

Editor's Note: The incorrect clues were published in Circuit Cellar*'s November 2013 issue, so last month's puzzle is being reprinted here with the correct clues. The answers will be available in the next issue and are posted at circuitcellar.com/crossword*



ACROSS

2. Temperature restrictions for laser cooling techniques [two words]
4. This signal sees the world through rose-colored glasses [two words]
5. k or kB [two words]
7. A half adder is made of an AND gate and one of these [two words]
14. Symbolized by "Np"
15. Two tunnel diodes used in high-speed gate circuits [two words]
17. Unidirectional antenna
18. Used as a light source in a cathode ray tube
19. All materials show this, but certain liquids display it more strongly than others [two words]

DOWN

1. This type generates sine waves [three words]
3. These digital circuits are built with several collector BJTs [three words]
6. Their electronic properties can be metallic or semiconducting [two words]
8. Uses diodes and transistors' logarithmic properties to compensate for nonlinearities and instabilities [two words]
9. Can be implied by the instruction's function [two words]
10. Used to measure a signal's standards
11. Semiconductor network sealed in a thin rectangular package
12. Process that limits peak signals [two words]
13. Condenses or enlarges an electric signal's dynamic range
15. Device under development by search engine giant
16. A very small robot

TEST YOUR EQ

Contributed by David Tweed

ANSWER 1

The first function, `test_pressure()`, converts the ADC reading to engineering units before making the threshold comparison. This is a direct, obvious way to implement such a function.

The second function, `test_pressure2()`, converts the threshold value to an equivalent ADC reading, so that the two can be compared directly.

The key difference is in performance. The first function requires that arithmetic be done on each reading before making the comparison. However, the calculations in the second function can all be performed at compile time, which means that the only run-time operation is the comparison itself.

ANSWER 2

First of all, note that there are really only four inputs and three unique outputs for this function, since input E is always 1 and outputs GHI are always 0. The only real outputs are F, plus the groups JK and LM.

Since the other 27 input combinations haven't been specified, we can take the output values associated with all of them as "don't care."

The output F is simply the inversion of input C. The output JK is high only when A is high or D is low. The output LM is high except when B is low and C is high. Therefore, the entire function can be realized with a total of five gates **(see Figure 1)**.

ANSWER 3

The original 10-Mbps Ethernet standard was jointly developed by Digital Equipment Corp. (DEC), Intel, and Xerox. It was released in November 1980, and was commonly referred to as "DIX Ethernet."

ANSWER 4

The multiple access with collision detection protocol that Ethernet uses was based on a radio protocol developed at the University of Hawaii. It was known as the "ALOHA protocol."

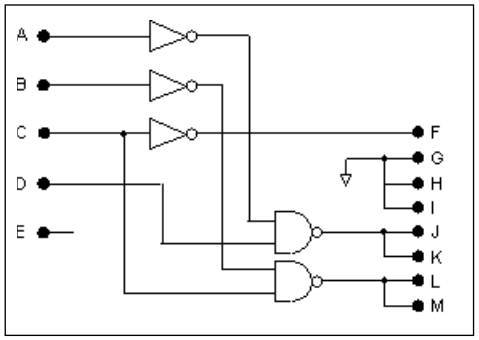


FIGURE 1
How many NAND gates would it take to implement Problem 2's transition table?

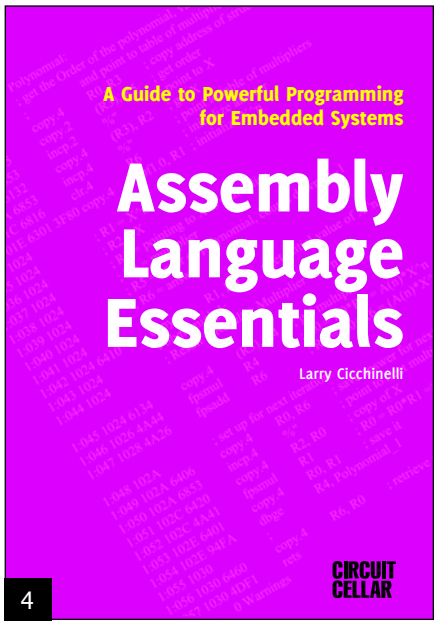
What's your EQ? The answers are posted at circuitcellar.com/eq. You can contact the quizmasters at eq@circuitcellar.com.

CC SHOP

1

2


3



4

1 CC GOLD

It's the ultimate archive for all things *Circuit Cellar*: project articles, tutorials, source code, project files, and more! The USB includes every *Circuit Cellar* issue in PDF format and article code from the beginning through your date of purchase. Keep your CC Gold archive up to date with a *Circuit Cellar* digital subscription. The archive is loaded with everything you need to know about: embedded applications and development, signal processing, wireless communications, programmable logic, embedded programming, measurement and sensors, analog techniques, data acquisition, Internet and connectivity, communications, robotics, graphics and video, and more.

Item #: CCGOLD • $250

2 CC 2012 ARCHIVE CD

A top-selling item, the CC 2012 Archive CD puts an entire year of embedded design solutions at your fingertips. With quick links to the all projects, source code, and schematics, CC archive CDs will become a staple in your prototyping artillery. Articles cover everything from energy-harvesting applications and environmental data logger construction, to developments in microelectromechanical systems and Arduino survival guides. The content on these CDs will help you master the evolving world of embedded design and development.

Item #: CD-017-CC2012
$34

3 EMBEDDED LINUX BOARD

Linux is available on many devices, but the OS's complexity and high development costs can be challenging at times. Not the case with this board! The beginner's course that accompanies this development board makes it an ideal kit for Linux experts and enthusiasts alike. Features include a two-layer board with readily available components, compatibility with standard debugging and programming software, 180 MHz, 32-MB RAM, and 64-MB swap.

Item #: EPS-ELNL-120026-91
$93

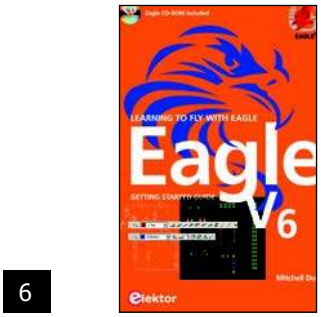
4 ASSEMBLY LANGUAGE ESSENTIALS

Looking to brush up your programming skills? Get back to to the basics with this matter-of-fact guide to Assembly language. Perfect for advancing students and academics, this book introduces you to a processor's most fundamental programming language. It includes essential terminology pertaining to higher-level programming, important algorithms that can be built into high-level language, a free downloadable Assembler program, and much more.

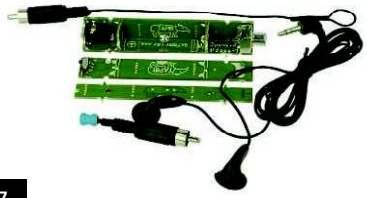
Author: Larry Cicchinelli
Item #: CC-BK-9780963013323
$47.50

CC SHOP

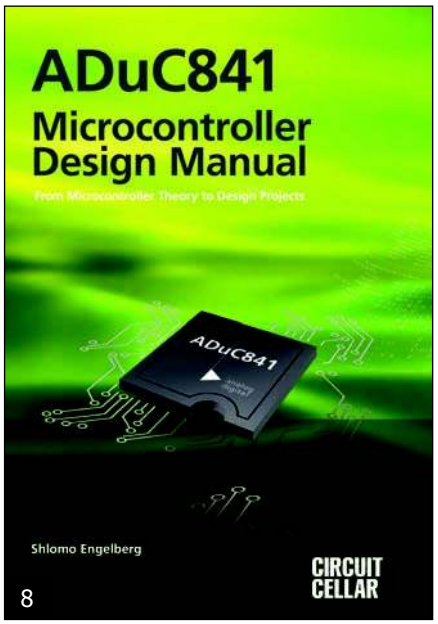
5



6



7



8

5 CC CLOUD TAG T-SHIRT

For a limited time, you can have an exclusive Circuit Cellar Tag Cloud T-shirt. Designed for electronics engineers, the T-shirt features "Circuit Cellar" on the front and an embedded electronics tag cloud on the back. It is 100% loose-fit cotton and currently available in large and extra-large sizes.

Item #: CC-SHIRT-001
$20

6 EAGLE V6 GETTING STARTED GUIDE

EAGLE is a user-friendly, powerful, and affordable software package for efficient PCB design. It can be used on most main computing platforms including: Microsoft Windows (XP, Vista, or Windows 7); Linux (based on kernel 2.6 or above), and Apple Mac OS X (Version 10.6 or higher). This book will benefit novices and professionals who are eager to learn about EAGLE or may be migrating from another CAD package. From schematic and layout editing tools to project completion, EAGLE V6 will help you achieve your PCB fabrication goals.

Item #: BK-ELNL-978-1-907920-20-2
$47.6

7 TAPIR

The TAPIR—which stands for Totally Archaic but Practical Interceptor of Radiation—is a fun and expensive kit that detects frequency levels in electronic devices. Once assembled, connect the headphones and antenna and switch it on. Move the TAPIR around any electrical device and you'll hear different noises, depending on the emitted field's type and frequency.

Item #: EPS-ELNL-120354-71
$21

8 ADUC841 MICROCONTROLLER DESIGN MANUAL

This manual presents a comprehensive guide to designing and programming with the Analog Devices. ADuC841 microcontroller and other microcontrollers in the 8051 family. It includes a set of introductory labs that detail how to use these microcontrollers' most standard features, and includes a set of more advanced labs, many of which make use of features available only on the ADuC841 microcontroller. The more advanced labs include several projects that introduce you to ADCs, DACs, and their applications. Other projects demonstrate some of the many ways you can use a microcontroller to solve practical problems. The Keil µVision4 IDE is introduced early on, and it is used throughout the book. This book is perfect for a university classroom setting or for independent study.

Author: Shlomo Engelberg
Item #: CC-BK-9780963013347
$35

Further information and ordering

www.cc-webshop.com

CONTACT US:
Circuit Cellar, Inc.
111 Founders Plaza, Suite 300
East Hartford, CT 06108
USA
Phone: 860.289.0800
Fax: 860.461.0450
E-mail: custservice@circuitcellar.com

IDEA BOX

the directory of PRODUCTS & SERVICES

AD FORMAT:

Advertisers must furnish digital files that meet our specifications (circuitcellar.com/mediakit).

ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT. E-mail adcopy@circuitcellar.com with your file.

For current rates, deadlines, and more information contact Peter Wostrel at 978.281.7708 or circuitcellar@smmarketing.us.

The Vendor Directory at circuitcellar.com/vendor is your guide to a variety of engineering products and services.

Can anyone deny that we're on the verge of major breakthroughs in the fields of embedded development, microcomputing, wireless communications, and robot design? Tech the Future is a section devoted to the ideas and stories of innovators who are developing the groundbreaking technologies of tomorrow.

Low-Cost SBCs Could Revolutionize Robotics Education

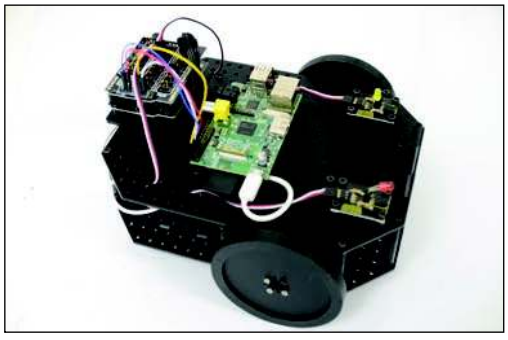
By Kyle Granat

Kyle Granat is a hardware engineer at Trossen Robotics, headquarted in Downers Grove, IL. Kyle graduated from Purdue University with a degree in Computer Engineering. Kyle, who lives in Valparaiso, IN, specializes in embedded system design and is dedicated to STEM education.

For my entire life, my mother has been a technology trainer for various educational institutions, so it's probably no surprise that I ended up as an engineer with a passion for STEM education. When I heard about the Raspberry Pi, a diminutive $25 computer, my thoughts immediately turned to creating low-cost mobile computing labs. These labs could be easily and quickly loaded with a variety of programming environments, walking students through a step-by-step curriculum to teach them about computer hardware and software.

However, my time in the robotics field has made me realize that this endeavor could be so much more than a traditional computer lab. By adding actuators and sensors, these low-cost SBCs could become fully fledged robotic platforms. Leveraging the common I²C protocol, adding chains of these sensors would be incredibly easy. The SBCs could even be paired with microcontrollers to add more functionality and introduce students to embedded design.

There are many ways to introduce students to programming robot-computers, but I believe that a web-based interface is ideal. By setting up each computer as a web server, students can easily access the interface for their robot directly though the computer itself, or remotely from any web-enabled device (e.g., a smartphone or tablet). Through a web browser, these devices provide a uniform interface for remote control and even programming robotic platforms.



A Raspberry Pi controls this robotic rover.

A server-side language (e.g., Python or PHP) can handle direct serial/I²C communications with actuators and sensors. It can also wrap more complicated robotic concepts into easily accessible functions. For example, the server-side language could handle PID and odometry control for a small rover, then provide the user functions such as "right, "left," and "forward" to move the robot. These functions could be accessed through an AJAX interface directly controlled through a web browser, enabling the robot to perform simple tasks.

This web-based approach is great for an educational environment, as students can systematically pull back programming layers to learn more. Beginning students would be able to string preprogrammed movements together to make the robot perform simple tasks. Each movement could then be dissected into more basic commands, teaching students how to make their own movements by combining, rearranging, and altering these commands.

By adding more complex commands, students can even introduce autonomous behaviors into their robotic platforms. Eventually, students can be given access to the HTML user interfaces and begin to alter and customize the user interface. This small superficial step can give students insight into what they can do, spurring them ahead into the next phase.

To see a video of a web-based interface, go to circuitcellar.com/ccmaterials

Students can start as end users of this robotic framework, but can eventually graduate to become its developers. By mapping different commands to different functions in the server side code, students can begin to understand the links between the web interface and the code that runs it.

Students will delve deeper into the server-side code, eventually directly controlling actuators and sensors. Once students begin to understand the electronics at a much more basic level, they will be able to improve this robotic infrastructure by adding more features and languages. While the Raspberry Pi is one of today's more popular SBCs, a variety of SBCs (e.g., the BeagleBone and the pcDuino) lend themselves nicely to building educational robotic platforms. As the cost of these platforms decreases, it becomes even more feasible for advanced students to recreate the experience on many platforms.

We're already seeing web-based interfaces (e.g., ArduinoPi and WebIOPi) lay down the beginnings of a web-based framework to interact with hardware on SBCs. As these frameworks evolve, and as the costs of hardware drops even further, I'm confident we'll see educational robotic platforms built by the open-source community.

FTDI Chip

AVAILABLE NOW

DISPLAY MADE EASY

EVE EMBEDDED VIDEO ENGINE

A REVOLUTIONARY SOLUTION ENABLING HI-QUALITY, HUMAN MACHINE INTERFACES AT A LOWER COST

DEVELOPMENT SYSTEMS FROM

$35

DISPLAY AUDIO TOUCH

WATCH EVE DEMOS ONLINE

FTDICHIP.COM